ABSTRACT

Title of Dissertation: Trust through Transparency: Towards Reliable AI for All

Mazda Moayeri
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Soheil Feizi
Department of Computer Science

Seemingly performant models can break down in unexpected and uneven ways, from image classifiers failing to recognize an otter out of water, to LLMs being nearly 3 times worse at recalling facts about Somalia than Sweden. In this dissertation, I'll detail interpretability techniques to scalably illuminate and efficiently intervene on discovered model deficiencies. First, I'll present evidence for pervasive reliance on spurious correlations by vision models, by way of carefully constructed benchmarks. Then, I'll automate these approaches, demonstrating the power of leveraging auxiliary models to more efficiently organize data, towards uncovering and articulating subsets where models struggle. Finally, I'll show how these same techniques can be applied to mitigate instances of real-world geographic disparities and even tackle sociotechnical challenges like artistic copyright infringement. In general, it can be difficult to trust what we do not fully understand, especially when unexpected failures arise. By scalably identifying failure modes before they cause harm, we enhance transparency around model abilities and limitations, thus better informing when models can be trusted to work reliably for all.

Trust through Transparency: Towards Reliable AI for All

by

Mazda Moayeri

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
January 31, 2025

Advisory Committee:
Professor Soheil Feizi, Chair/Advisor
Professor Tom Goldstein
Professor Hal Daume III
Professor Katie Shilton
Professor Behtash Babadi

© Copyright by
Mazda Moayeri
2025

Dedication

For my parents – Babanader and Elhamie – to whom I owe everything.

Acknowledgments

First and foremost, I'd like to thank my advisor, Soheil Feizi. Without you, I would not be here. You gave me a chance when I was just a 21 year old kid coming from a tiny liberal arts college who's mind was wrapped up in deep learning for music. After an isolated first semester in the midst of a pandemic when I didn't really know where to begin, you gave me the push I needed to shed my fear of failure and get started. Once the ball was rolling, you were my biggest advocate and constantly reminded me to believe in myself more. I will always find confidence in your belief in me, and I will always appreciate all the ways in which you have helped me. The time has flown. I'm grateful for it all, and I can't wait to see all the more you will achieve.

I'd like to thank my lab mates, especially all those who I've collaborated with: Phil, Yogesh, Sahil, Kiarash, Keivan, Wenxiao, Samyadeep, Sriram, Priyatham, Atoosa, Arman, Parsa and Sumit. Sahil, thank you for choosing to work with me when I was still very early in my journey; I'm glad it was productive for us both. Kiarash and Keivan, it brought me great joy to collaborate with you on some of your first papers in graduate school; you both did great jobs. I'd also like to acknowledge my other fellow terps who's work and lives I've had the pleasure to hear about: Gaurang, Alex, Chenghao, Mehrdad, Aviva, Aya, Neha, Shweta, Shoumik, Vinu. I am especially grateful for Aya and Neha's encouragement, from my first presentation to the lab to my job applications – it all meant the world to me. Lastly, shoutout to UMD CS's fearless intramural soccer team. We may have never clinched the championship, but our journey from losing every

game to winning multiple playoff matches by the end was a blast. Given that our team was called the Neural Network, the fact we won any games at all defied the odds at least a bit.

I am also very grateful to my collaborators outside of UMD. Thomas Fel, your work is insane, and the fact that I am on your radar is a great honor. Thank you for inviting me to speak at Brown and for all the collaborations we've had. To all my industry collaborators – Mark, Diane, Mike, and Maziar from Meta, and Vibhav, Neel, Vidhisha, Varun, Besa, and Safoora at MSR – thank you so much; my internships were instrumental in my personal and professional development. Special shoutout to the interns on my MSR team: Sid, Alessandro, and Tash. Who would've thought we'd have as much fun as we did that summer!

I'd like to thank the ARCS foundation for their recognition and support, as well as UMD for various awards during my PhD. Moreover, I am very thankful for the administrative staff – Tom, Migo, Jodie, Sharron – without whom UMD would fall apart.

Finally, my heart goes out to my best friends – Mikey, Jill, Ava, Rowan, Dylan, Lizzy – who made DC a home for me, made sure I fed myself during deadlines (even bringing me food if they suspected I was forgetting), reminded me to celebrate, and filled my life with so many memories. And to Ari, words can't express how happy you've made me. Without you all, I'd be a shell of who I am today. My heart is forever yours.

And of course, my parents!!! You have made me who I am today. Thank you for all your advice, support, countless sacrifices, and patience. I must've spent hours complaining about work, and you've always reminded me to keep my head up. You've done everything you possibly could to help me, and I will forever be in debt to you. You inspire me more than you know. I was not expecting to end up close to home after undergrad, and I am so grateful that the universe brought me back here. I will always cherish this time. This degree belongs to all of us. I love you.

Trust through Transparency: Towards Reliable AI for All

by

Mazda Moayeri

Doctor of Philosophy Dissertation
September 2020 - January 2025



DEPARTMENT OF
COMPUTER SCIENCE

Table of Contents

Dedication ii

Acknowledgements iii

Title Page v

Table of Contents vi

List of Tables xi

List of Figures xiii

Chapter 1: Introduction 1
1.1 Motivation . . . 1
1.2 Overview and Contributions . . . 5

I Case Study: Spurious Correlations in Vision Models 10

Chapter 2: A Comprehensive Analysis of Image Classifier Sensitivity to Foregrounds, Backgrounds, and Attributes 12
2.1 Introduction . . . 13
2.2 Review of Literature . . . 16
2.2.1 Related Datasets . . . 16
2.2.2 Interpretability Methods . . . 18
2.3 RIVAL10 . . . 19
2.3.1 Overview . . . 19
2.3.2 Data Collection . . . 20
2.4 Models . . . 21
2.5 Foreground and Background Sensitivity . . . 23
2.5.1 Noise Analysis . . . 23
2.5.2 Empirical Observations . . . 25
2.5.3 Removing Backgrounds Entirely . . . 28
2.5.4 Saliency Alignment . . . 28
2.6 Conclusion . . . 30

Chapter 3: Hard ImageNet: Segmentations for Objects with Strong Spurious Cues 31

3.1 Introduction . . . 32
3.2 Related work . . . 34
3.3 Hard ImageNet Dataset . . . 35
3.3.1 Overview . . . 35
3.3.2 Collection Procedure . . . 37
3.4 Benchmarking Spurious Feature Reliance . . . 38
3.4.1 Ablation . . . 39
3.4.2 Relative Foreground Sensitivity . . . 41
3.4.3 Saliency Alignment . . . 42
3.5 Hard ImageNet Objects are Smaller and Less Centered . . . 44
3.6 Ranking Images by Spurious Feature Presence . . . 47
3.7 Improving Models with Hard ImageNet Annotations . . . 48
3.8 Conclusion . . . 50

Chapter 4: Explicit Tradeoffs between Adversarial and Natural Distributional Robustness 51
4.1 Introduction . . . 52
4.2 Review of Literature . . . 55
4.3 Theoretical Analysis on Linear models . . . 58
4.4 Empirical Evidence . . . 64
4.4.1 AT hurts Natural Distributional Robustness *only* when Spurious Correlations are broken . . . 64
4.4.2 Reduced Core Sensitivity, and Difference in the Effect of ℓ_2 & ℓ_∞ Adversarial Training . . . 66
4.4.3 Adversarial Training Increases Background Reliance in Synthetic Datasets 67
4.4.4 Reverse Effect: Presence of Spurious Correlations Can *Improve* Adversarial Robustness . . . 70

II Scalable Discovery and Mitigation of Deficiencies 72

Chapter 5: Sample Efficient Detection and Classification of Adversarial Attacks via Self-Supervised Embeddings 74
5.1 Introduction . . . 75
5.2 Prior Works . . . 79
5.2.1 Adversarial Attacks and Defenses . . . 79
5.2.2 Self-Supervised Encoders . . . 80
5.3 SimCLR distance as proxy for Perceptibility . . . 82
5.4 Methods . . . 83
5.4.1 General SimCat Framework . . . 83
5.4.2 SimCat Variants . . . 85
5.5 Detection and Classification . . . 86
5.5.1 Evasion Attacks . . . 86
5.5.2 Poisoning Attacks . . . 89
5.6 Generalization to Unseen Models . . . 91
5.7 Hardening SimCat to an Adaptive Attack . . . 92

5.7.1 Attack Formulation . . . 92
5.7.2 Adversarial Training . . . 93
5.7.3 Results . . . 94
5.8 Conclusion . . . 96

Chapter 6: Text-To-Concept (and Back) via Cross-Model Alignment 97
6.1 Introduction . . . 98
6.2 Review of Literature . . . 102
6.3 Model Alignment . . . 104
6.4 Text to Concept . . . 106
6.4.1 Method Details . . . 107
6.4.2 Zero-Shot Classification . . . 109
6.5 Additional Applications of Text-to-Concept . . . 112
6.5.1 Concept-Bottleneck Networks for Free . . . 112
6.5.2 Concept-Based Dataset Summarization and Distribution Shift Diagnosis . . . 114
6.5.3 Concept Logic for Image Retrieval . . . 115
6.6 Concept-to-Text . . . 117
6.7 Conclusion . . . 118

Chapter 7: Unearthing Skill-Level Insights for Understanding Trade-Offs of Foundation Models 120
7.1 Introduction . . . 121
7.2 Constructing skill-slices using Model-Generated Rationales . . . 125
7.2.1 Skill Extraction via Rationale Parsing . . . 126
7.2.2 Curating Skill-Slices across Benchmarks . . . 127
7.2.3 Automated Validation of Skill Relevance . . . 127
7.3 Analyzing Foundation Models with Skill-slices . . . 129
7.3.1 Finer-grained Insights from Existing Evaluation Instances . . . 130
7.3.2 Generalization of Skill-Slice Insights . . . 132
7.4 Corroborating Slice Accuracies with Probing Questions . . . 134
7.5 Retrieving Evaluation instances for Custom Query Skills . . . 136
7.6 Review of Literature . . . 138
7.7 Discussion and Future Work . . . 140

Chapter 8: Prioritizing Interpretability in Failure Mode Extraction 141
8.1 Summary . . . 141

Chapter 9: Spuriosity Rankings: Sorting Data to Measure and Mitigate Biases 146
9.1 Introduction . . . 147
9.2 Review of Literature . . . 151
9.3 Discovering Spurious Features in ImageNet and Beyond . . . 153
9.3.1 5000 Reasons Deep Models Use to Perform ImageNet Classification . . . 153
9.3.2 Spurious Feature Discovery *without* Adversarial Training . . . 155
9.3.3 Is a Human in the Loop Necessary? . . . 157
9.4 Measuring and Mitigating Biases with Spuriosity Rankings . . . 158

9.4.1 Spuriosity Rankings: Organizing Image Data via Robust Neural Features 158
9.4.2 Measuring Bias: Computing Spurious Gaps . . . 159
9.4.3 Mitigating Bias: Closing Spurious Gaps . . . 162
9.5 Additional Application: Flagging and Fixing Labeling Errors . . . 164
9.6 Conclusion . . . 167

III Applications to Sociotechnical Challenges 168

Chapter 10: WORLDBENCH: Quantifying Geographic Disparities in LLM Factual Recall 170
10.1 Introduction . . . 171
10.2 Related Work . . . 174
10.3 Methods: WORLDBENCH . . . 176
10.3.1 Data . . . 176
10.3.2 Language Model Evaluation . . . 178
10.4 Evaluation Suite . . . 181
10.5 Results: Pervasive and Consistent Geographic Disparities . . . 182
10.5.1 Large disparities across Regions and Income groups . . . 182
10.5.2 Error nearly triples between some countries . . . 183
10.5.3 Consistent disparities across LLMs and indicators . . . 183
10.5.4 Ordering of Regions and Income groups by error are consistent per-LLM and per-Indicator: Lowest error is with Western and high income groups . 186
10.6 Noteworthy Observations . . . 188
10.6.1 Citation Hallucination . . . 188
10.6.2 Are some LLMs already out of date? . . . 190
10.6.3 What kinds of countries experience high error rates? . . . 191
10.7 Limitations . . . 192
10.8 Conclusion . . . 193

Chapter 11: Embracing Diversity: Interpretable Zero-shot classification beyond one vector per class 194
11.1 Introduction . . . 196
11.2 Review of Literature . . . 199
11.3 Motivation . . . 201
11.3.1 A single vector inadequately represents diverse classes . . . 202
11.3.2 A path forward: VLMs can recognize diversity with relevant attributes . . 203
11.4 Method . . . 203
11.4.1 Attribute Inference Along Many Axes of Diversity . . . 204
11.4.2 Nonlinear Prediction Consolidation . . . 205
11.5 Analysis . . . 207
11.5.1 Consistent Gains Across Diverse Datasets, Particularly for the Hardest Classes and Subpopulations . . . 207
11.5.2 Enhanced Interpretability at No Cost to Performance . . . 211
11.6 Ablations . . . 214
11.6.1 Scaling with the Many Axes of Diversity . . . 215

11.6.2 Tunable Trade-off between Accuracy Overall and On Worst Classes . . . 215
11.7 Limitations . . . 217
11.8 Conclusion . . . 218
11.9 Appendix: A deeper understanding . . . 219
11.9.1 When can we cram an entire class in one vector, and when can we not? . 219
11.9.2 One final Trade-off . . . 223

Chapter 12: Rethinking Artistic Copyright Infringements in the Era of Text-to-Image Generative Models 226
12.1 Introduction . . . 227
12.2 Related Works . . . 232
12.3 Motivation: Image-wise similarity may be limited for Style Copying . . . 234
12.4 Reformulating Artistic Style Copying as Classification over Image Sets . . . 235
12.4.1 DeepMatch: Black-Box Detector . . . 236
12.4.2 Interpretable Artistic Signatures . . . 238
12.4.3 TagMatch: Interpretable and Attributable Style Detection . . . 241
12.5 `ArtSavant`: A Practical Tool for Concerned Artists . . . 242
12.5.1 Analysis with `ArtSavant`: Quantifying Style Copying of 372 Prolific Artists . . . 244
12.6 Conclusion . . . 247
12.7 Appendix . . . 248
12.7.1 A Brief History of Copyright Law: Situating our work . . . 248
12.7.2 A nuance in artistic style infringements: Existing Artists can have very similar styles . . . 251
12.7.3 Human Validation of Style Copying . . . 256

Chapter 13: Looking ahead 259

Bibliography 260

List of Tables

3.1 Final layer retraining improves faithful learning on Hard ImageNet. Results shown for entire benchmark under two different training approaches: i) Core Risk Minimization (CoRM) via *random background noising* and *saliency regularization*, and ii) deep feature reweighting (DFR) using a *spurious-balanced training subset*. We also report results for the combination of the two approaches and ordinary finetuning (as a baseline) under two architectures. Relative Foreground Sensitivity (RFS) is evaluated under two ℓ_∞ noise levels, indicated by σ. Saliency refers to *saliency alignment* as measured by intersection over union (IoU). 49

5.1 Performance of SimCat for detection and classification using few training samples on SVHN (top) and ImageNet (bottom). In parenthesis, we denote the improvement gained by using SimCat compared to a baseline using supervised embeddings. For ImageNet, the detector is trained and evaluated over all eight attack types at once, and classification is done over all eight attack types. 87

5.2 Results for SimCat detection and classification of poisoning attacks on STL10 using ten samples per poisoning. Asterisk indicates removing FC poisons. 89

5.3 Poison defense via SimCat detection. PSR is poison success rate. SimCat model is trained on CP and BP jointly. Ens. SimCat trains a CP detector and a BP detector separately, then filters any samples that are detected as poison by either detector. Both defenses only use *ten* samples of CP and BP poisons each. 90

5.4 Generalization of SimCat detectors to unseen threat models. First row shows accuracy of detector trained on a single attack evaluated on all other attacks. Other rows contain accuracy of a SimCat detector trained on the union of all other attacks. The second column indicates the number of samples per threat model used in training. 91

5.5 Ablation study on SimCat AT. Accuracy refers to attack agnostic ImageNet detection from Section 5.1. Robustness is measured as the percent of test adversarial samples that can be adaptively attacked with PGD-ℓ_2, $\epsilon = 2.0$ to be misdetected as clean. AT is done for 20 epochs. Aug refers to augmenting both the original clean and adversarial samples - this is distinct from AT+Aug, where only the clean samples are augmented to balance out the addition of adaptive adversarial attacks to the SimCat training set. 95

6.1 Percent of captions for decoded class vectors deemed relevant to images in the class by human annotators. 117

10.1 Global development indicators in WORLDBENCH, each defined and maintained by the World Bank. . . . 173
10.2 (**Left**) Correlation between per-country mean absolute relative error and individual indicator values. (**Right**) Per-indicator, correlation between per-country mean absolute relative error and normalized standard deviation of responses obtained over five trials. . . . 191

11.1 Zero-shot classification on datasets with known variation types for CLIP with a ViT-B/16 encoder. States averages results over the two categorizations of MIT States data, while Hierarchical averages results over four Breeds datasets. We observe similar results for BLIP-2. . . . 209
11.2 Zero-shot classification performance on geographically diverse images from DollarStreet and GeoDE using CLIP with a ViT-B/16 encoder. We see similar results for BLIP-2. . . . 210
11.3 Zero-shot classification performance on finer-grained **held-out** datasets without attributes, using CLIP with a ViT-B/16 encoder. We observe similar results for BLIP-2. Our method effectively generalizes to new settings without tuning the set of queries for attribute inference. . . . 211

List of Figures

1.1 Models can be "right for the wrong reasons", leading to failure modes that go undetected. In Part I, manually constructed benchmarks that go beyond single-class labels enable closer study of spurious correlations in vision models. 2

1.2 Part II shows how auxiliary models enable scaling up analyses from Part I to a much wider set of potential failure modes; e.g., skills extracted from an auxiliary model's rationales lead to enhanced transparency around tradeoffs between top foundation models (see Chapter 7). 3

1.3 Undetected failure modes can have adverse and disparate societal impact. Part III focuses on a few such implications: catching geographic disparities in LLMs, curtailing them in VLMs, and examining artistic copyright infringements of text-to-image models. 5

2.1 Examples where background noise degrades performance of highly accurate models more than foreground noise. Gaussian ℓ_∞ noise with standard deviation $\sigma = 0.24$ shown. Probabilities are averaged over ten trials. While these examples are atypical, we observe that they are surprisingly prevalent, and model design can affect the degree to which such cases arise. 13

2.2 Accuracy under noise in foregrounds and backgrounds, averaged over multiple noise levels. Marker size is proportional to parameter count. Models with higher relative foreground sensitivity lie further from the diagonal. 15

2.3 (**Left**): Correlations between attributes in the training split. (**Right**): Class-wise means of attribute vectors in the training split. 19

2.4 Accuracy under noise in foreground (**left**) and background (**middle**) at various noise levels. Models are grouped by architecture and training procedure, with a curve corresponding to the average over all models in a group. (**Right**): RFS by group. 22

2.5 Relative foreground sensitivity per instance for four classes and five models of roughly equal size. (**Top**): Histogram of $iRFS$; positive denotes greater foreground sensitivity. (**Bottom**): Scatter; top left indicates high relative foreground sensitivity. Across models, ships and birds have low foreground sensitivity, often being more sensitive to noise in the background than the foreground. 24

2.6 Controlled ablation studies. Average RFS over all noise levels presented for brevity. (**Left**): Increasing patch size in ViTs decreases relative foreground sensitivity. (**Right**) Robust models are much less relatively sensitive to foregrounds, but ϵ used in adversarial training does not affect RFS much. 26

2.7 Instances of images with low saliency alignment, highlighting spurious features of water for ships, and branches and bird feeders for birds. (**Left**): Spurious features leading to misclassification (in red). (**Right**): Other instances of spurious features. . . . 26
2.8 Model accuracy on images with backgrounds ablated via graying. The right plot shows accuracies for models finetuned on the images with backgrounds ablated. Only transformers can fit a linear layer on the features of background ablated images without compromising performance. . . . 27
2.9 Alignment of binarized saliency maps with object segmentation masks, measured by intersection over union (IOU). Averaged over models (**left**) and object classes (**right**). . . . 29

3.1 *Hard ImageNet* classes. Object segmentation masks are overlaid in purple. . . . 32
3.2 Agreement of individual segmentations with final averaged segmentation for Hard ImageNet validation split. . . . 37
3.3 Accuracy under ablation. Accuracy drops much less when Hard ImageNet objects are ablated than when RIVAL10 objects are ablated. Example ablations for class *sunglasses* shown left. . . . 40
3.4 Relative Foreground Sensitivity (RFS) for DeiT and ResNet, off the shelf (left) and finetuned (right). RFS for Hard ImageNet is significantly lower, and often negative, indicating greater sensitivity to background noise than foreground noise. 42
3.5 GradCAMs of Hard ImageNet instances with poor saliency alignment for finetuned ResNet50 (left three) and finetuned DeiT (right two). Spurious features of *gymnast, dogs, hockey stick and ice, legs* and *keys* for each class respectively are highlighted. . . . 42
3.6 Alignment of GradCAM saliencies to object masks. Low alignment indicates more saliency is placed in the background. Across all models, saliency alignment is lowest for Hard ImageNet. . . . 43
3.7 Average object position for Hard ImageNet and RIVAL10. Hard ImageNet objects occupy less space and are often not centered, unlike RIVAL10 (and most other ImageNet) objects. . . . 44
3.8 Hard ImageNet objects occupy less space in images than RIVAL10 objects both before and after standard augmentation. . . . 45
3.9 Hard ImageNet objects are often substantially cropped out by the standard ImageNet test time augmentation. . . . 45
3.10 Examples of high (left) and low (right) spurious feature presence, based on Hard ImageNet rankings. Accuracies shown are for a standardly trained ResNet50. . . . 46

4.1 Snapshot of empirical evidence using *RIVAL10, Salient ImageNet-1M, ImageNet-9, Waterbirds,* and *ObjectNet* benchmarks. Results averaged over ResNet18 and ResNet50. **Adversarial training, especially under ℓ_2 norm, reduces (increases) sensitivity to core (spurious) features. The increased reliance on spurious features leads to worse distributional robustness**. . . . 53

4.2 Reliance of the adversarially trained model on spurious features as measured by NFS (see Equation (4.5)) for different choices of ℓ_p norm and different values of adversarial budget ϵ and spurious correlation parameter η. 62
4.3 Norm fraction over spurious features (NFS) with an ℓ_∞ constrained adversary as a function of the scale of the spurious features and perturbation budget ϵ. 63
4.4 Effect of reliance on spurious features on distributional robustness. Each figures compares two models, one using only the core features, and another using all of the features (denoted by "core" and "total" respectively). For the ℓ_1 and ℓ_2 norms, the scale of the spurious features is 1 while for the ℓ_∞ norm, the scale is set to 3. 64
4.5 OOD accuracy vs standard ImageNet accuracy for adversarially trained ResNets. ImageNet-C accuracy is closely tied to standard accuracy, but for ObjectNet, where spurious correlations are broken, performance drop is more severe than a linear relation with standard accuracy would entail. 65
4.6 Noise-based evaluation of model sensitivity to foreground (RFS on RIVAL10) or core (RCS on Salient ImageNet-1M) regions. Lower values entail greater sensitivity to spurious regions. 66
4.7 Background Gap (difference in accuracies on ImageNet-9 subsets MIXED-SAME and MIXED-RAND). The drop in accuracy due to background cross-class swapping (MIXED-RAND) causes larger drops in accuracy for robust models, especially ℓ_2 adversarially trained models. 68
4.8 Accuracies on Waterbirds subsets where spurious correlation is intact (majority group; e.g. land birds on land backgrounds) and where it is broken (minority group; e.g. land birds on water backgrounds). The drop in performance due to breaking the background correlation grows larger for robust models, especially ℓ_2 adversarially trained models. 69
4.9 Injecting spurious feature improves adversarial robustness. 70

5.1 Perceptibility of adversarial attacks relative to the ℓ_2 distance between the original and perturbed image in the SimCLR embedding space. Each point in the scatter plot refers to the average distance between adversarial examples within a single threat model under one of three bounds. Correlation is $r = 0.854$. 81
5.2 The distributions of ℓ_2 distances between adversarial examples and clean images in pixel space (left) and SimCLR embedding space (right). Note the different scales; the distributions across threat models are much more uniform when using SimCLR embeddings. 82
5.3 SimCat classification accuracy over eight diverse attack types. The classifier is fit on just 25 samples per class. Overall classification accuracy is 40.7%, an 11.8% increase over baseline. 88
5.4 Generalizability of SimCat detectors to unseen threat models. Each detector is trained on 100 samples from a single threat model, and evaluated on all other models. 92
5.5 SimCat robustness to adaptive PGD-ℓ_2 over epochs of adversarial training with varied values of the hyperparameter β, which controls the momentum updates. Higher values of β lead to more emphasis on the linear classifiers solved in later epochs. Adversarial training is unstable without momentum ($\beta = \infty$). 95

6.1 Overview. After *aligning* the representation space of a given image encoder to a CLIP image encoder, we can compare aligned representations of images to concept vectors obtained *directly from text* (typically, an example set of data is required to obtain each concept vector; our method is example-free, i.e. O(1) w.r.t. data collection). Using a GPT-based CLIP decoder, we can map arbitrary vectors in representation space to text. Our method yields efficient **interpretability**: we only train *one linear layer*. 98
6.2 Qualitative validation of text-to-concept. ImageNet classes are sorted by the average cosine similarity of the CLIP embedding for "in a tree" to the *linearly aligned* Dino ResNet representations of images within each class. Highest ranked classes indeed often appear in a tree, as is evident by the most similar instances. The least similar instances appropriately do not contain the concept. 101
6.3 Heatmap of retained accuracy. The value in row r and column c is the average of retained accuracy when doing alignment from all models in group r to all models in group c. Note that "Sup" refers to supervised while "SS" refers to self-supervised models. All models except CLIPs are trained on ImageNet. . . . 105
6.4 Text-to-concept can encode finer-grain concepts, like combinations of concepts (*"red food"*) or textures (*"polka-dotted"*). 108
6.5 The zero-shot capabilities of CLIP can extend to off-the-shelf vision encoders via alignment based text-to-concept. (**Left**) Models trained on ImageNet can recognize coarse categorizations of ImageNet classes, despite never explicitly being taught them. (**Right**) Off-the-shelf models remain strong zero-shot classifiers even when images are out of distribution. In some cases, they surprisingly surpass the accuracy of the CLIP vision encoder whose jointly-trained text encoder was used to embed each class vector. 110
6.6 Edge cases for zero-shot classification. (**Left**) Models struggle with OCR. (**Right**) Models can recognize some primitive concepts by name. Same legend as figure 6.5. 111
6.7 Example inference for a Concept Bottleneck Model (CBM) obtained via training a linear layer on zero-shot concepts. Since logits in the CBM are linear functions of concept scores, we can precisely quantify the contribution of concepts to each logit. 113
6.8 Concept similarities can reveal distribution shifts, like in ObjectNet, where photos are taken within people's homes. 114
6.9 Images retrieved based on similarity of their representation to multiple text-to-concept vectors. Retrieved images satisfy the multiple conditions listed above each image. $\sim$ denotes 'not'. Concept logic with text-to-concept enables searching over images, while allowing the use of any vision encoder to represent them. . 116
6.10 Overview of our concept logic image retrieval technique. Instead of directly searching for images of *orange cats that are not indoors*, we decompose the query into concept constraints of varying degree. Our method grants the querier more control in their search, and also circumvents limitations of CLIP's text encoder when processing negations and longer queries. 116

7.1 We leverage model-generated rationales to extract the skills relevant to any evaluation instance. Inspecting accuracy along *skill-slices* (instances drawn across benchmarks involving the same skill) surfaces fine-grained insights otherwise obfuscated by aggregate accuracy. . . . 122
7.2 **(left)** Sample GPT-4o generated rationale: each skill is listed under multiple names of *cascaded granularity*, and localized to a specific step and concluding claim. **(right)** Annotated skills can be verified independently with a second model or human. We include randomly sampled negative skills and multiple verifiers to assure the quality of the verification, as detailed in 7.2.3. . . . 125
7.3 (**left**) Post-hoc verification shows GPT-4o-annotated skills are relevant, and that automatic verifiers are reliable, as they admit low rates of false positives (marking a randomly sampled negative skill as relevant). (**right**) Rate that GPT-4o-annotated skills are marked as relevant, separated by if GPT-4o correctly answered the underlying evaluation instance (blue) or not (orange). Empirically, annotated skills have high relevancy rates *even* when the annotator incorrectly answers the question. . . . 128
7.4 Skill-slices shed insight on how models evolve over new releases. For GPT and Claude models, skills related to law see the largest increases in accuracy, while Gemini models improve most in performing math and science skills. . . . 129
7.5 Unique strengths (**top**) and weaknesses (**bottom**) of GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet, relative to one another. We present skills where each model's slice accuracy is highest / lowest (respectively) relative to the average of the other two model accuracies. † denotes significance ($p < 0.01$) via Wilcoxon signed-rank test; †† means both gaps vs. the other models are significant. . . . 131
7.6 (**left**) A simple routing scheme that assigns each instance to the model with best accuracies for the skills relevant to that instance can lead to noticeable accuracy gains. (**right**) Accuracies over skill-slices can generalize to unseen data. In fact, skill-wise accuracies computed over multimodal benchmarks correlate strongly with accuracies obtained over a language-only benchmark. . . . 133
7.7 (**left**) Skills who's slices yield lowest model accuracy have the highest rates of inconsistency when probed. Each point corresponds to one model's consistency rate for a single skill. (**right**) Probe questions are generated by reframing rationale steps associated with a skill of interest. . . . 135
7.8 Retrieving instances based on a query skill is far more effective when utilizing skill annotations than using direct representations of each instance, as done in prior work. . . . 137

8.1 Overview of PRIME. Prior methods cluster inputs in an uninterpretable activation space, leading to semantically misaligned failure mode descriptions. PRIME utilizes an auxiliary model to obtain faithful attributes, before efficiently combining them to extract specific and accurate descriptions. . . . 143
8.2 Although appearance of tags "hang", "black", and "branch" individually lowers model's accuracy, when all of them appear in the images, model's accuracy drops from 86.23% to 41.88%. . . . 145

9.1 Spuriosity rankings sort images based on the presence of spurious cues. The rankings reveal hidden spurious features that models rely on, like *rippling water, human hands,* and *flame in the dark* (right; top to bottom), and uncover minority subpopulations where spurious cues are absent (left). 147
9.2 Overview of our framework. Per class, we sort images by *spuriosity*, which we proxy via the activation of neural features (i.e. nodes in the penultimate layer of a trained classifier) that focus on spurious cues. After sorting, measuring model bias to the discovered spurious features is as simple as computing the drop in accuracy across high and low spuriosity images. This bias can be mitigated efficiently by tuning the classification head on low spuriosity images. 148
9.3 Examples of the visualizations we use to identify the focus of a robust neural feature. Here, feature 595 focuses on flowers, making it core for *Daisy* and spurious for *Monarch Butterfly*. 153
9.4 With only linear layer retraining, we detect designated spurious attributes like background in the Waterbirds benchmark (left), as well as new spurious dependencies such as racial bias in Celeb-A Hair Classification (right). Presented above are highest activating images (top row), corresponding heatmaps (middle), and lowest activating images (bottom) for spurious features observed for the classes *waterbird, landbird, blond hair,* and *brown hair* (left to right). 156
9.5 Our lightweight extension of [1] also reveals spurious features in *non*-spurious correlation benchmarks, such as UTKFace gender and race classification. Spurious cues are absent in the lowest activating images (bottom row). Thus, sorting by feature activation uncovers minority subpopulations. 157
9.6 All models are biased: accuracy is 10 to 30% lower when spurious cues are absent compared to when they are present (**left**). However, bias varies far more significantly across classes than across models (**right**), emphasizing that bias due to relying on spurious cues is crucially class-dependent. 160
9.7 Class-wise spurious gaps are strongly correlated across models. While spuriosity rankings are empirically computed with the help of a single ℓ_2 adversarially trained (AT) ResNet, diverse models show similar sensitivities to the detected spurious features. 161
9.8 (**Left**) Finetuning on low spuriosity images closes reduces spurious gap to below 5% at little cost of validation accuracy. (**Right**) Closing spurious gap removes bias to images with high spurious ranks; that is, regardless of spuriosity, the model performs with roughly the same accuracy. 162
9.9 (**Left**) Training images with highest spuriosity for the classes with the most negative spurious gaps. These images often contain multiple objects, at times more prominently showing an object from a class different than the image's label (e.g. *car*, *monitor* or *keyboard*, *spider*). (**Right**) Using neural activation maps for neural features annotated as *core* (i.e. not spurious) for the class may serve as an automated way to crop out the conflicting objects. 165

10.1 WORLDBENCH leverages World Bank data to assess the ability of large language models (LLMs) to recall factual information about specific countries. Above, we plot the absolute relative error per country, averaged over 11 global development statistics queried to 20 state of the art open source and private LLMs. WORLDBENCH reveals significant geographic disparities in LLM factual recall. 171
10.2 Overview of WORLDBENCH. Our benchmark provides a manner to quantify the performance of large language models (LLMs) on a per-country basis. We disentangle data collection from evaluation by utilizing the World Bank's data bank, which contains statistics (called indicators) pertaining to numerous diverse aspects of global development. Crucially, the data is available for nearly all countries and is updated year to year. With WORLDBENCH, one can flexibly select specific statistics of interest, and dynamically re-evaluate models as time passes to see if they remain up to date. In this work, we uncover substantial geographic disparities in LLM performance for a wide range of models released by industry leaders, revealing the inequities pervasive across state of the art LLMs. 172
10.3 World Bank categorizations of countries into 7 regions (left) and 4 income groups (right). 177
10.4 Standard pipeline for extracting numeric answers from LLMs. Each question is defined by a query (i.e. Before asking a language model a question, we prompt it with a base instruction and example. Then, we automatically parse the raw output to obtain a numeric value which can be compared to the groundtruth data. 178
10.5 Language models exhibit disparate performance for countries from different regions and income groups. Error rates are lower for western and high income countries. Mean absolute relative error rate per region and income group reported over all 11 queries and 20 language models studied. We note that the best performing LLMs have much lower error rates than the averages presented above (see figure 10.7). 182
10.6 Error rates can vary significantly across countries, with some countries experiencing nearly $3\times$ higher absolute relative error than others. Strikingly, all of the 15 countries with the lowest error rates fall in the high income category, while all of the 15 countries with the highest error rates fall in the low income category. . . 183
10.7 Performance of 20 LLMs averaged over 11 indicators from WORLDBENCH. We present the absolute relative error (left), as well as disparities across regions (middle) and income groups (right). For disparities, the blue dashed lines correspond to the disparity incurred using a random categorization of countries (into 7 groups for Regions and 4 for Income groups), averaged over ten trials. Observed disparities far exceed the amount expected for a random categorization of countries across nearly all LLMs. 184
10.8 Error rates and disparities per indicator, averaged over LLMs. For disparities, the blue dashed lines correspond to the disparity incurred using a random categorization of countries (into 7 groups for Regions and 4 for Income groups), averaged over 10 trials. 186

10.9 The order of regions and income groups by absolute relative error is largely consistent per LLM (top) and per indicator (bottom). For both LLMs and indicators, the regions with the lowest errors are most frequently North America and Europe & Central Asia, while the regions with the highest error are most frequently Sub-Saharan Africa and East Asia & Pacific. For Income groups, error nearly always increases as income decreases. 187
10.10 In addition to hallucinating false answers, we also observe LLMs to occassionally hallucinate *citations*. Above, a few examples of hallucination citation are shown. 188
10.11 Error rate of LLM outputs compared to year from which groundtruth is extracted. Many models show the lowest error rate when their outputs are compared to groundtruths from 2021, indicating that models may already be slightly out of date. 190

11.1 (**Left**) Instances of a class can appear in many diverse ways, like the pears above. Using one vector (the classname embedding) to represent the whole class results in disparate performance, particularly for atypical instances. (**Middle**) To address this issue, we infer attributes and embed multiple vectors, reducing disparities and enhancing interpretability. (**Right**) Our method scales better than prior works as we include more attributes (Section 11.6.1), enabling us to account for the many ways in which diversity can arise. 195
11.2 We test models on datasets that provide groundtruth **attributes** (shown in bold) annotating hierarchies, diverse states, and real-world shifts (e.g., Rojas et al. [2] labels the income level and country of origin of each image, towards promoting AI models that reduce bias) within a class. We find that standard zero-shot accuracy ('Base Acc.' above) drops significantly when certain attributes are present, namely when the attribute manifests in visual differences from what the model considers 'typical' for the class. We design our method to improve performance on these 'atypical' instances. 196
11.3 The average precision (AP) of a classname embedding is often much lower than the average precision of a subpopulation (i.e. classname with attribute) embedding. Subpopulations that see large increases in AP by including the attribute tend to be atypical. We design our method to improve accuracy on these diverse subpopulations, by inferring and explicitly accounting for them. 202
11.4 An *Arctic* `fox` can more closely resemble a typical `wolf` than a typical `fox`. Standard zero-shot classification using one vector per class (the classname embedding) is ill suited for this case. We address this issue by nonlinearly consolidating similarities to *multiple* vectors per class that explicitly encode the diverse subpopulations within the class. See section 11.4.2 for full explanation. 206
11.5 Instances where our method corrects mistakes of the standard approach. The attributes used in inference also serve as faithful fine-grained explanations. Notably, these samples are atypical, suggesting that inspecting samples where our method and standard classification disagree can enable automatic surfacing of atypical cases, towards better understanding the task at hand. 213

11.6 Accuracy, overall and for the worst classes, as new types of attributes are added. Performance for our consolidation scheme continuously improves, while it saturates or deteriorates for others. 214
11.7 **Right:** As k decreases, first, accuracy overall and on the worst classes both increase. Then, overall accuracy begins to decrease while accuracy on the worst classes continues to improve. Thus, we can control this trade-off via k. **Left:** λ, the continuous analog of k, allows for greater control. 216
11.8 Arctic Fox bias is amplified in zero-shot classifier vs. to supervised linear probes. 222
11.9 Averaging subpopulation vectors before computing similarity to an image embedding proves to be another way to trade overall accuracy for accuracy on the worst classes. That is, when we first compute similarity to each subpopulation and then average, we obtain higher overall accuracy but lower accuracy on the worst classes, compared to when we first average subpopulation vectors and then compute the similarity to the average vector. 224

12.1 Our primary contribution is an interpretable and quantitative framework, rooted in legal scholarship, for arguing style infringement from the perspective of classification. Given artworks by Canaletto, our tool, `ArtSavant`, automatically identifies a unique style and recognizes said style in generated art, summarizing its findings in an easy to understand yet quantitative report. 229
12.2 We define artistic style as a set of elements (or signature) that appear frequently over a body of work, and reduce the problem of style copy detection to classification of *sets* of images to artists. (**left**) We offer proof-of-concept via two ways to recognize artistic styles over image *set*, including a novel inherently interpretable and attributable tag-based method. (**right**) In an empirical study of 372 prolific artists, we find generative models potentially copy artistic styles for 20.2% of these artists. 231
12.3 Example generations from Stable Diffusion 2 when prompted to produce specific paintings by Vincent Van Gogh, along with the histogram of similarities between the generated image and corresponding real image. Even for a famous artist like Van Gogh, generative models rarely produce near-exact duplicates. However, Van Gogh's *style* appears consistently, even when similarity is low. 235
12.4 DeepMatch on held-out real art: 89.3% of artists can be recognized. The remaining 10.7% of artists have very similar styles to other artists: e.g., Palma Il Giovane's work differs marginally from other Italian renaissance painters. 237
12.5 Example atomic tags assigned via our proposed CLIP-based zero-shot method. We perform selective multilabel classification along various aspects of art (e.g. medium, colors, shapes, etc), so that atomic tags span diverse categories. Details in section 12.4.2. 238
12.6 Composing atomic tags results in more unique tags, towards artistic *tag signatures*. 240
12.7 `ArtSavant` flow. We design our tool with a concerned artist in mind, who wishes to quickly investigate the degree to which they may be at risk of style copying by generative models. 243
12.8 DeepMatch on generated art. In red: the fraction of artists with their styles recognized in at least half of their respective generated images. 244

12.9 Examples of applying TagMatch to generated images. TagMatch is inherently interpretable with respect to tags, as each inference comes with the exact set of tags that are (i) shared between the sets of test art and art from the predicted artist, and (ii) used to predict the artist. 246
12.10Examples of artists who's styles were not recognized by DeepMatch (i.e. less than half of their held-out works were predicted to the artist). Each panel shows an example work from (left) the unrecognized artist and (right) the artist that is incorrectly predicted most frequently over works from the unrecognized artist. We see that artists can use very similar, at times arguably indistinguishable, styles. 252
12.11We verify the stricter criterion of *unprecedented similarity* by holding out the real artist with highest similarity to a given artist, and checking if the held-out real artist's works are flagged as potential style copying by DeepMatch. (**left**) We observe only three artists where the most similar held-out artist has their work flagged as a style match, and in all cases, when generated images are flagged, the match confidence of the generated images exceeds that of the held-out real artist's works (i.e., **the generated images flagged by our method reflect *unprecedented similarity* to the given artist's style**). (**right**) Inspecting the flagged held-out artists further show that style copying is very nuanced, as artists take inspiration from one another, and as such, they may already have very similar styles. While we always observe unprecedented similarity, a potential solution to style copying may be for generative models to ensure that they do not copy any more than what already exists; that is, they may exhibit some copying, but no more than for which precedent already exists. 254
12.12Samples visualizations from our human evaluation. In the real evaluation, panel titles are removed and order is randomized. The first case demonstrates an instance where the human annotator viewed the generated art as more stylistically similar to the plaintiff's art than the most similar other artist, suggesting that Renoir's unique style is at risk of copying. The latter two cases show different ways in which the threshold of style copying may not be met: when the generated art is not similar to the plaintiff, or when the generated art is not sufficiently *uniquely* similar to the plaintiff. 258

Chapter 1: Introduction

1.1 Motivation

Deep learning-based artificial intelligence (AI) has grown rapidly and consistently, signaling the advent of an era where its use as a general-purpose technology becomes ubiquitous. Despite demonstrating impressive performance on a wide array of tasks, the excitement felt within the AI community is not mirrored by the general public: a 2023 poll conducted by Pew Research found 52% of Americans were more concerned than excited about the increased use of artificial intelligence, while only 10% felt more excited than concerned [3]. One cause for concern around the widespread adoption of AI is reliability. Models have been shown to fail in a number of diverse ways: imperceptible perturbations can cause vision models to produce completely irrelevant outputs [4]; accurate models can experience significant drops in performance when spurious correlations present in the training data are broken [5, 6, 7]; large language models (LLMs) can confidently and convincingly generate false information [8]. Further, failure modes can disparately affect some demographics more than others, posing serious fairness issues [9].

While any technology has limitations, ensuring the reliability of AI poses unique challenges. A traditional approach to verify some technology is safe to use consists of first contemplating the most likely ways in which the technology may fail, and then testing the technology in each scenario. However, because AI can be used in so many settings, it may also fail in more

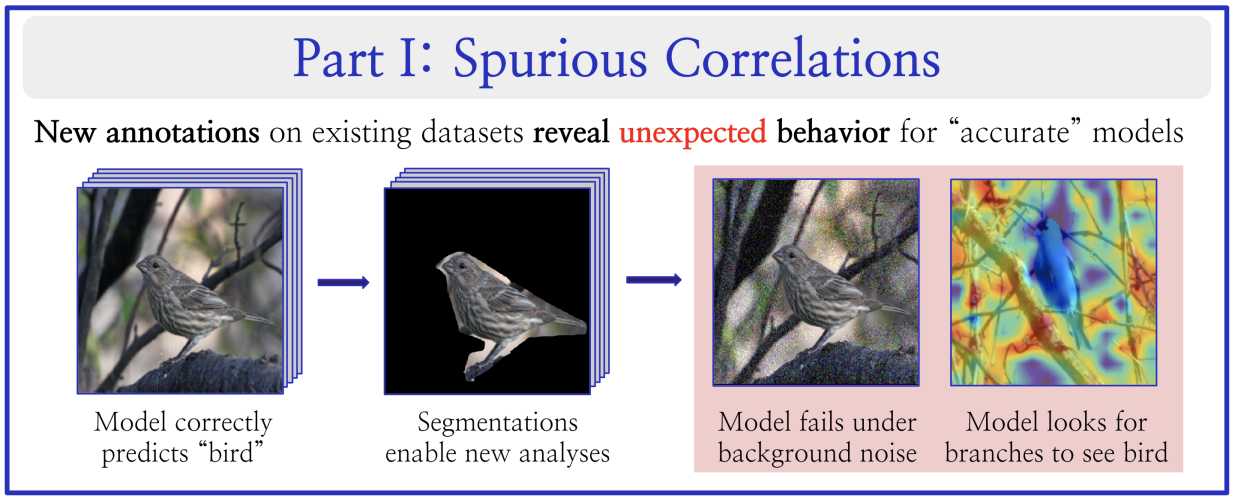


Figure 1.1: Models can be "right for the wrong reasons", leading to failure modes that go undetected. In Part I, manually constructed benchmarks that go beyond single-class labels enable closer study of spurious correlations in vision models.

settings than one can enumerate apriori. As the AI community continues to trend towards creating general purpose foundation models [10] which have countless use cases, the prospect of manually verifying each risk becomes less and less feasible. Even when we are aware of a potential failure mode, it can be expensive to verify the model's robustness to the failure, as one would typically need to gather more data that specifically encapsulate the scenario of interest. Needless to say, manually gathering new data for every scenario one wishes to test would incur a prohibitively high cost. Moreover, AI models can fail in ways that humans would not expect. Thus, safety concerns are compounded by the fact that these moments of unreliability can be difficult to predict, and as such, often go unnoticed until they cause harm.

Consider the example of verifying that a simple image classifier does not rely on spurious correlations[1]. First, since training and validation sets are typically drawn from identical distributions, the correlations present in the training set are likely to be preserved in the validation

[1]Spurious correlations are patterns that often (but crucially, not always) co-occur with a true signal of interest. For instance, a classifier may learn to rely on the presence of grass to recognize a cow because most of the cows seen during training are in pastures. This model may then fail to recognize cows native to the island of Corsica, which are often pictured on beaches [11].

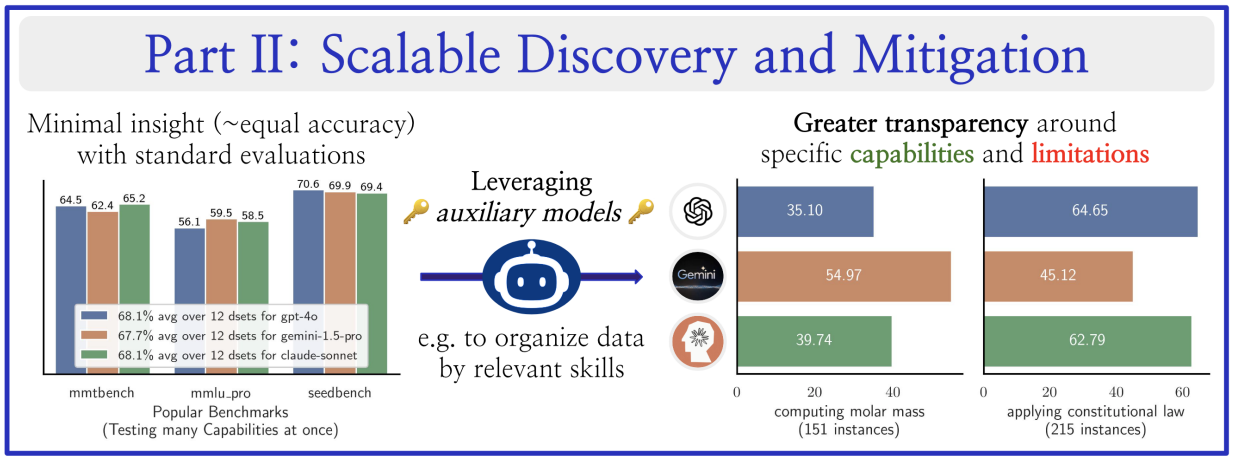


Figure 1.2: Part II shows how auxiliary models enable scaling up analyses from Part I to a much wider set of potential failure modes; e.g., skills extracted from an auxiliary model's rationales lead to enhanced transparency around tradeoffs between top foundation models (see Chapter 7).

set. Thus, even if the model does use spurious features, the validation accuracy may very well remain high, providing a false sense of security. To truly assess the reliability of the classifier, one would need to obtain a new set of data where spurious correlations are broken, which is expensive. Moreover, one would need to know what spurious correlations are relevant before finding data that break them, a task made challenging by the black-box nature of most AI models and amplified by the ever-growing size of models and training datasets alike.

Emphasizing this last point, a bottleneck in building reliable AI and fostering trust amongst users is a lack of transparency. For one, the inner-workings of models are uninterpretable, making it challenging to detect improper reasoning that may lead to errors. More broadly, standard evaluation practices often leave ambiguity to the exact limits of a model's capabilities: average accuracy over a standard benchmark can be insufficient to inform a user if the model will work for their specific case, nor will it provide intuition to (on a qualitative level) the types of instances where the model is more likely to struggle.

In this dissertation, I present methods to enhance transparency around how AI models make

decisions and the cases in which they may be unreliable, towards systematically uncovering and remedying model limitations. I argue that to ensure the reliability of an AI system, we should first derive a concise and global understanding of how the system operates; namely, when it fails and when it can be relied upon to work as expected. To do so requires searching over the space of potential failure modes, which is particularly vast for AI because of the technology's versatility. Thus, scalability is a central desiderata of the proposed methods. Towards this end, I demonstrate the potential of leveraging some models in an auxiliary fashion to aid in automatically auditing a much wider set of failure modes than manual approaches would allow. Finally, I illustrate how reliability concerns can manifest as fairness issues, adversely affecting certain populations of users much more than others, before applying techniques developed across my dissertation to tackle some of these instances. Often, the solutions I propose replace existing workflows with more interpretable analogs that explicitly account for corner cases. Indeed, for one of the sociotechnical applications we investigate – artistic copyright infringement – transparency is not just a goal, but a requirement, as judges, lawyers, and juries cannot incorporate the insights from our method if they are not interpretably conveyed.

In summary, this dissertation offers multiple perspectives to reliability challenges in AI, from close inspection of spurious correlations in an academic setting to constructing transparent-by-design systems to account for and mitigate disparities that emerge for real-world geographic shifts. I present evidence for the existence of limitations that go undetected with standard evaluation practices, and offer scalable augmentations to evaluations to more interpretably diagnose model weaknesses, as well as strengths. Then, I demonstrate how these insights can enable mitigations, often in efficient manners. For example, after a surfacing a subset of data where a model struggles, improving performance on that subset can at times be as simple as tuning the model

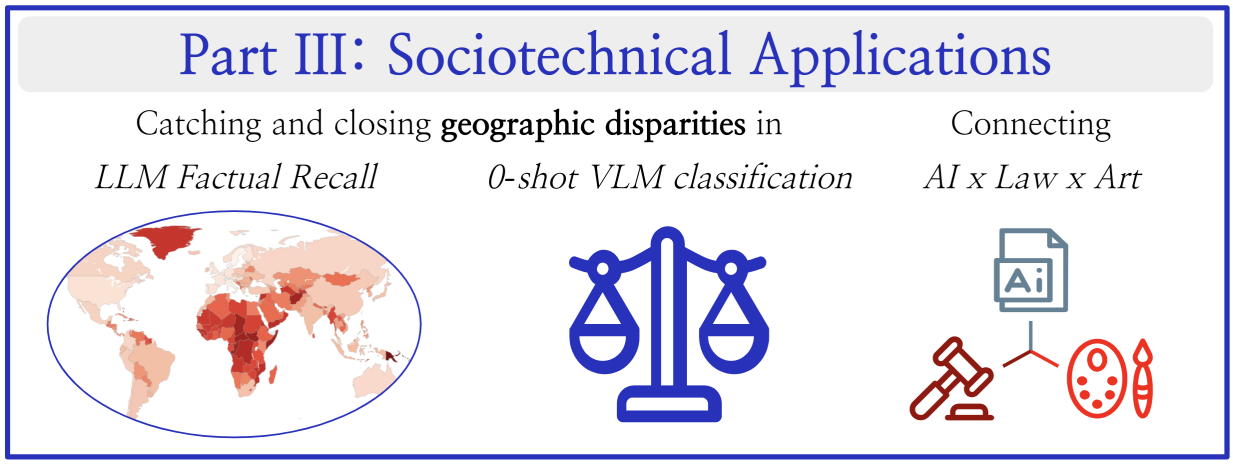


Figure 1.3: Undetected failure modes can have adverse and disparate societal impact. Part III focuses on a few such implications: catching geographic disparities in LLMs, curtailing them in VLMs, and examining artistic copyright infringements of text-to-image models.

on that same subset. As a whole, ensuring reliability requires a continual effort, which persists as models evolve and new applications emerge. Scalability and transparency are crucial tenants of the solutions I propose, as (i) the space of use-cases (and consequently corner cases) for AI is too wide to cover with manual methods, and (ii) without clarity around model limitations, the occurrence of unexpected failures will denigrate user trust. Indeed, these points are all the more important when applying a sociotechnical lens to AI reliability, where overlooked corner cases translates to real people being adversely affected. While AI need not be perfectly understood by all who use it, enhancing transparency around when it can be relied upon will foster greater trust for broader swaths of society, allowing AI and its potential benefits to reach a wider audience.

1.2 Overview and Contributions

I now provide a more detailed overview of the technical contributions made in this dissertation, along with information regarding how this document is organized.

In Part I, I will take a close look at spurious correlations in vision models. This case study

epitomizes the phenomenon that seemingly performant models can be "right for the wrong reasons", resulting in unexpected failure modes that overall accuracy obfuscates. I will also show that different model limitations can be at odds such that tackling one issue may amplify another, which suggests that model reliability will not have a one-size-fits-all solution. Notably, these insights utilize new benchmarks that go beyond single class-label supervision (e.g. object segmentations or labels for attributes), so to provide richer signal than accuracy alone. While these benchmarks can be illuminating, they are often expensive to construct. To expand these approaches, in Part II, I will present a number of works that allow for more scalable analyses of model limitations. The common thread in these works is collaborating with auxiliary models to better understand a model of interest. To name a few, I propose methods to more efficiently interpret the activation space of vision models, understand skill-level strengths and weaknesses of foundation models, and even mitigate the spurious correlation issue detailed in Part I. Finally, in Part III, I turn to problems with more direct societal impact. Specifically, I measure geographic disparities in LLMs, mitigate them in zero-shot vision-language model (VLM)-based image classification, and present a legally-grounded quantitative framework for characterizing artistic styles and detecting potential style infringements by text-to-image generative models.

Each chapter following this one will correspond to a separate conference main proceedings publication, presented in slightly abridged form. Mainly, appendices are removed; readers can refer to the corresponding orginal paper for complete details. In total, this dissertation presents findings from 10 papers I led over my PhD (3 at NeurIPS, 2 each at ICLR and FAccT, and 1 each at CVPR, ICML, and ICCV), along with a short chapter for an ICLR paper where I advised as a later author. Below, I provide brief summaries of each chapter, along with their key contributions.

Part I: Case Study on Spurious Correlations in Vision Models

- **Chapter 2** reveals an unexpected behavior of image classifiers when operating under localized noise; namely, background noise can degrade performance more than foreground noise in some instances. This issue has roots in spurious correlation reliance, which we study for a vast set of models using multiple novel analyses and a new dataset. **Key contributions:** RIVAL10 dataset, noise-based *relative foreground sensitivity* metric, saliency alignment technique for finding relied upon spurious cues.

- **Chapter 3** identifies a subset of classes where image classifiers heavily rely on spurious cues, to the extent that they still predict that an object is present even when it has been removed entirely. Abnormally high reliance on spurious cues for these classes is shown to occur across diverse model types, and properties of these classes are studied. **Key contributions:** Hard ImageNet dataset, object ablation analysis.

- **Chapter 4** presents a surprising tradeoff between two notions of robustness, with extensive theoretical and empirical support. **Key contributions:** theory-based understanding of an unintended consequence of adversarial training, curation of a benchmark suite for measuring spurious correlation reliance.

Part II: Scalable Discovery and Mitigation of Deficiencies, demonstrating the power of leveraging auxiliary models.

- **Chapter 5** shows adversarial examples are linearly separable in the embedding space of a frozen self-supervised model (SimCLR [12]). **Key contributions:** efficient detection and classification of adversarial attacks.

- **Chapter 6** extends CLIP [13] text encoders to arbitrary vision models by training just a

linear layer, greatly reducing the cost of interpreting the latent spaces of vision models. **Key contributions**: Text2Concept, Concept2Text, cross-model alignment.

- **Chapter 7** uses model-generated rationales to extract skill-level strengths and weaknesses from existing foundation model evaluations. **Key contributions:** SkillIndex dataset, skill-based (training-free) routing, skill-based retrieval of evaluation instances.

- **Chapter 8** presents an efficient manner to obtain more coherent and faithful descriptions of image classifier failure modes, utilizing a pretrained image attribute tagger. **Key contributions:** PRIME, automatic metrics for assessing quality of failure mode descriptions.

- **Chapter 9** uses neuron-level interpretability techniques on an adversarially trained surrogate model to uncover spurious cues for image classifiers and organize data into subsets of high or low *spuriosity* (i.e. if the spurious cue is present or absent). This enables measurement of model bias to spurious cues using natural images (as opposed to prior noise or ablation based approaches) without needing to collect new data. We also mitigate these biases efficiently at no cost to overall accuracy. **Key contributions:** spuriosity rankings for ImageNet [14] classes, empirical evidence for widespread class-dependent biases over dozens of diverse models, technique for discovering relevant cues for arbitrary classification datasets, measurement and mitigation of biases, method for finding and fixing label noise.

Part III: Applications to Sociotechnical Challenges

- **Chapter 10** leverages data from the WorldBank to reveal that geographic disparities in factual recall are pervasive in LLMs. **Key contributions:** WORLDBENCH dataset.

- **Chapter 11** tackles geographic disparities in zero-shot VLM-based image classification via collaboration with LLMs and a novel diversity-aware classification paradigm. The resultant system has enhanced transparency and improved accuracy, driven by gains on the most challenging subsets of data, including for images from regions and income-levels that typically see highest error rates. **Key contributions:** novel interpretable and fairer zero-shot classification scheme, principled tradeoff analysis of accuracy and fairness.

- **Chapter 12** presents a legally-grounded quantitative approach to define unique artistic styles, demonstrate if they exist for an artist, and detect if the style is replicated in generative art. Large emphasis on interpretability and connection to existing legal scholarship. **Key contributions:** ArtSavant, novel interpretable and attributable tag-based art style classification approach, diversity aware zero-shot art tagging, empirical analysis of copying for state-of-the-art text-to-image generative models.

Part I: Case Study: Spurious Correlations in Vision Models

We begin by taking a close look at spurious correlations in image classifiers. Image classification is fundamental task in computer vision, where an input image must be mapped to one of a predefined set of classes. This task has been studied extensively, and image classification or object recognition is just a sub-task to the more complex reasoning over images that modern models are designed to do. Thus, models, especially deep-learning based models, can typically appear strong in image classification, achieving high accuracy on numerous benchmarks. Nonetheless, our analyses indicate that these models can be 'right for the wrong reasons'. That is, models may rely on spurious signals that may not always be reliable. In this chapter, we present evidence for the reliance of vision models on spurious cues, even for the simple task of image classification. Namely, we develop new datasets with annotations that go beyond single-class labels, along with novel noise-, saliency-, and ablation-based analyses, to quantify and understand the nature of spurious correlation reliance. Our analyses point to lead to surprising findings, including with respect to how spurious correlation reliance is affected by measures to address other reliability concerns. Specifically, while adversarial training improves a model's robustness to adversarial distribution shifts, we find that it results in greater reliance on spurious cues, and consequently, a reduced robustness to natural distribution shifts that break spurious correlations. Thus, reliability issues can come in distinct forms, and the interplay between them is not always predictable. These findings underscore the unexpected nature of model limitations, and the need for efficient methods – beyond standard evalution practices – to traverse the space of potential failures and shed light on reliability concerns before they cause harm.

Chapter 2: A Comprehensive Analysis of Image Classifier Sensitivity to Foregrounds, Backgrounds, and Attributes

This chapter details a paper presented as an oral at CVPR '22, co-authored by Phillip Pope, Yogesh Balaji, and Soheil Feizi [15].

While datasets with single-label supervision have propelled rapid advances in image classification, additional annotations are necessary in order to quantitatively assess how models make predictions. To this end, for a subset of ImageNet samples, we collect segmentation masks for the entire object and 18 informative attributes. We call this dataset RIVAL10 (**RI**ch **V**isual **A**ttributes with **L**ocalization), consisting of roughly $26k$ instances over 10 classes. Using RIVAL10, we evaluate the sensitivity of a broad set of models to noise corruptions in foregrounds, backgrounds and attributes. In our analysis, we consider diverse state-of-the-art architectures (ResNets, Transformers) and training procedures (CLIP, SimCLR, DeiT, Adversarial Training). We find that, somewhat surprisingly, in ResNets, adversarial training makes models more sensitive to the background compared to foreground than standard training. Similarly, contrastively-trained models also have lower relative foreground sensitivity in both transformers and ResNets. Lastly, we observe intriguing adaptive abilities of transformers to increase relative foreground sensitivity as corruption level increases. Using saliency methods, we automatically discover spurious features that drive the background sensitivity of models and assess alignment of saliency maps with fore-

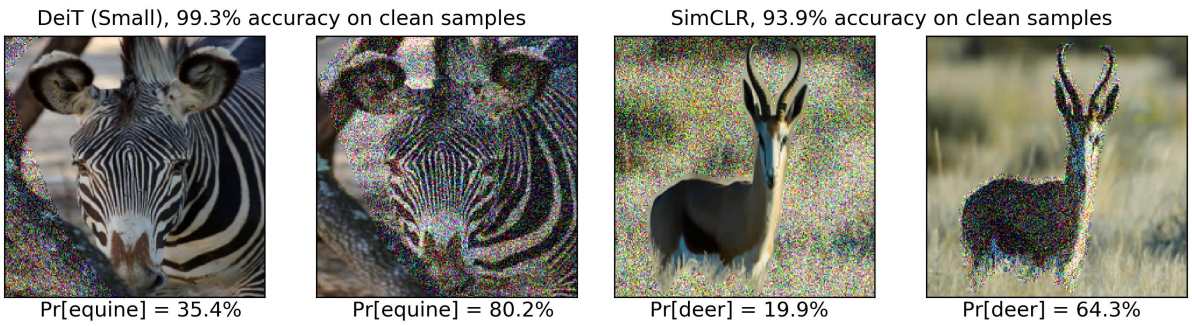


Figure 2.1: Examples where background noise degrades performance of highly accurate models more than foreground noise. Gaussian ℓ_∞ noise with standard deviation $\sigma = 0.24$ shown. Probabilities are averaged over ten trials. While these examples are atypical, we observe that they are surprisingly prevalent, and model design can affect the degree to which such cases arise.

grounds. Finally, we quantitatively study the attribution problem for neural features by comparing feature saliency with ground-truth localization of semantic attributes.

2.1 Introduction

Large scale benchmark datasets like ImageNet[14] that were constructed with single class label annotation have propelled rapid advances in the image classification task[16, 17, 18, 19]. Over the last decade, several network architectures and training procedures were proposed to yield very high classification accuracies [16, 18, 20, 21]. However, methods to interpret these model predictions and to diagnose undesirable behaviors are fairly limited. One of the most popular class of approaches are saliency methods[22, 23, 24, 25] that use model gradients to produce a saliency map corresponding to the most influential input regions that yielded the resulting prediction. However, these methods are qualitative, require human supervision, and can be noisy, thus making their judgements potentially unreliable when made in isolation of other supporting analysis.

In this paper, we argue that to obtain a proper understanding of how specific input re-

gions impact the prediction, we need additional ground truth annotations beyond a single class label. To this end, we introduce a novel dataset, RIVAL10, whose samples include **RI**ch **V**isual **A**ttributions with **L**ocalization. RIVAL10 consists of images from 20 categories of ImageNet-1k [14], with a total of $26k$ high resolution images organized into 10 classes, matching those of CIFAR10 [26]. The main contribution of our dataset is instance wise labels for 18 informative visual attributes, as well as segmentation masks for each attribute and the entire object. We present our dataset as a general resource for understanding models trained on ImageNet. We then provide a study of the sensitivity of a diverse set of models to foregrounds, backgrounds, and attributes.

Our study of background and foreground model sensitivity is motivated by some counter-intuitive model behaviors on images whose background and foreground regions were corrupted with Gaussian noise: Figure 2.1 shows instances where highly accurate models have performance degraded much more due to the background noise than the foreground noise. While this is not the norm (i.e. models are more sensitive to foregrounds on average), the existence of these examples warrants greater investigation, as they expose a stark difference in how deep models and humans perform object recognition. Quantifying the degree to which different architectures and training procedures admit these examples can shed new insight on how models incorporate foreground and background information.

To this end, we conduct a *noise analysis* that leverages object segmentation masks to quantitatively assess model sensitivity to foregrounds relative to backgrounds. We proxy sensitivity to a region by observing model performance under corruption of that region. We propose a normalized metric, *relative foreground sensitivity* (RFS), to compare models with various general noise robustness. A high RFS value indicates that the model uses foreground features in its inferences more than background ones since corrupting them result in higher performance degradation.

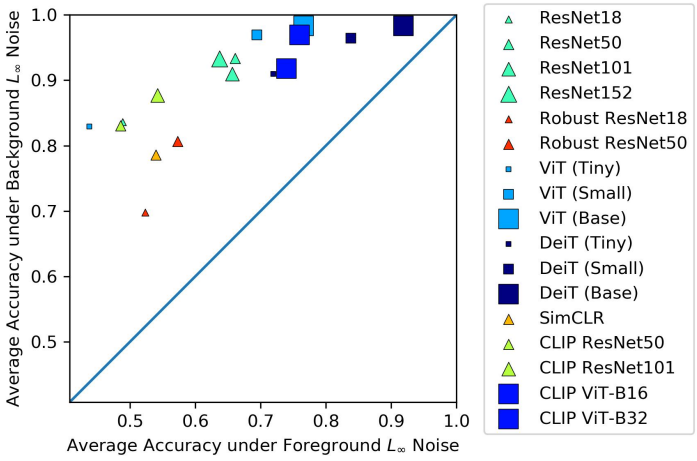


Figure 2.2: Accuracy under noise in foregrounds and backgrounds, averaged over multiple noise levels. Marker size is proportional to parameter count. Models with higher relative foreground sensitivity lie further from the diagonal.

In Figure 2.2, we see different architectures and training procedures lead to variations in both general noise robustness (projection onto the main diagonal) and relative foreground sensitivity (normalized distance orthogonal to the diagonal). Notably, we find that adversarially training ResNets significantly reduces RFS, surprisingly suggesting that robust ResNet models make greater use of background information. We also observe contrastive training to reduce RFS, and transformers to uniquely be able to adjust RFS across noise levels, reducing their sensitivity to backgrounds as corruption level increases. Lastly, we find object classes strongly affect RFS across models.

We couple our noise analysis with saliency methods to add a second perspective of model sensitivity to different input regions. Using RIVAL10 segmentations, we can quantitatively assess the alignment of saliency maps to foregrounds. We also show how we can discover spurious background features by sorting images based on the saliency alignment scores. We observe that performance trends that our noise analysis reveals are not captured using qualitative saliency methods alone, suggesting our noise analysis can provide new insights on model sensitivity to

foregrounds and backgrounds.

Lastly, we utilize RIVAL10 attribute segmentations to systematically investigate the generalizability of neural feature attribution: for a neural feature (i.e., a neuron in the penultimate layer of the network) that achieves the highest intersection-over-union (IOU) score with a specific attribute mask on top-k images within a class, how the IOU scores of that neural feature behave on other samples in that class. For some class-attribute pairs (e.g. dog, floppy-ears), we indeed observe generalizability of neural feature attributions, in the sense that test set IOUs are also high.

In summary, we present a novel dataset with rich annotations of object and attribute segmentation masks that can be used for a myriad of applications including model interpretability. We then present a study involving three quantitative methods to analyze the sensitivity of models to different regions in inputs. We hope the RIVAL10 dataset will help study failure modes of current deep classifiers and pave the way for building more reliable models in the future.

2.2 Review of Literature

2.2.1 Related Datasets

Prior to the rise of deep learning, a number of works studied attribute classification, leading to the construction of datasets such as Animals with Attributes [27] and aPASCAL VOC 2008 [28] (adding annotations to [29]). [30] published CUB 200, a fine-grained classification dataset of bird species with object segmentations and part *localizations* in the form of single coordinates, as opposed to segmentation masks like in RIVAL10. Finally, [31] collected object attributes on a small-scale subset of ImageNet. More recently, [32] publish a large-scale object attribute dataset on a subset of ImageNet. The Celeb-A dataset [33] contains attribution with applications to

generative modeling, but limited utility for general representation learning since it only contains face images. The broader dataset Visual Attributes in the Wild (VAW)[34] provides large-scale in the wild attribute annotations for 250k object instances.

Many datasets aim to stress test models to reveal limitations. [35] introduces ImageNet variants under diverse corruption types, including Gaussian noise. [36] adds two more ImageNet variants that include challenging natural samples and out of distribution samples, on which top models see massive accuracy drops. Models evaluated on [37] similarly see large drops, though this dataset differs in that it is strictly a test set. Other works introduce synthetic datasets to assess spatial biases [38] or background reliance of classifiers, such as [39] and [40], which perform some variation of swapping or altering foregrounds and backgrounds. Though similar, these works differ in objective and technical contribution to ours. [40] focuses on developing a novel distributionally robust optimization procedure. [39] emphasizes designing a multitude of test datasets through creative editing of foreground and background regions to serve as a general benchmark to evaluate models. In contrast, our work presents a novel method to analyze foreground sensitivity, and demonstrates its utility by applying it to a breadth of cutting-edge architectures and training paradigms, leading to *model-specific* observations. Further, our RIVAL10 dataset is significantly larger and richer in annotation.

Recently, [1] uses saliency maps and feature visualization in a semi-automated process to identify deep neural nodes corresponding to core or spurious features for an object of a given class, resulting in a large-scale dataset with segmentations corresponding to salient features. However, annotation of the segmented regions are limited to just labeling them as 'core' or 'spurious'.

2.2.2 Interpretability Methods

A number of methods have been proposed to interpret model predictions, such as saliency or class activation maps[22], influence functions [41], and surrogate white box models[42, 43]. However, saliency maps have been found to be noisy and influence functions are fragile [44, 45]. Some methods seek to interpret the function of a neural node via synthesizing inputs that maximize its activation [46, 47, 48], though these methods are limited when non-adversarially robust models are used[49], and offer qualitative insights.

A motivation behind the development of interpretability methods is to work towards addressing the 'shortcut learning' issue, where models rely on easy-to-learn features that lead to high performance on training sets, but poor generalization in other settings. [5] discusses this at length, recommending the development and usage of challenging datasets whose inputs are out-of-distribution with respect to standard benchmarks. RIVAL10's rich annotations open the door to the construction of many challenge datasets, in which shortcuts are broken via swapping backgrounds, foregrounds, and *attributes* (examples in Appendix).

Other constructive works aimed to reduce the reliance of deep models on spurious features appeal to counterfactual data generation [50, 51, 52], often appealing to disentangled representations or explicit annotations to break correlations of texture, shapes, colors, and backgrounds. Further, [53] found that removing spurious features can in fact hurt accuracy and disproportionately affect groups. Thus, the notion that spurious features are always harmful is incomplete, and a closer look is required to ground discussions regarding the shortcut learning issue. Lastly, [54] provides theoretical context for stress testing models to discern causal factors.

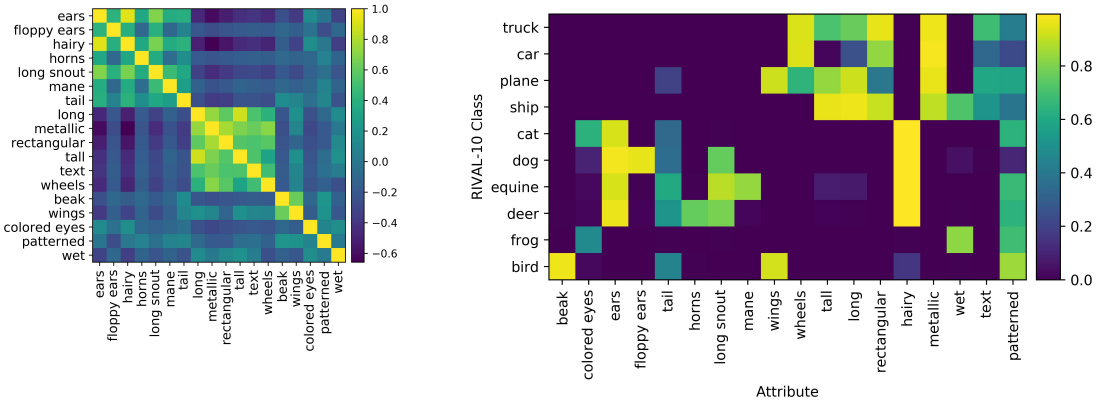


Figure 2.3: (**Left**): Correlations between attributes in the training split. (**Right**): Class-wise means of attribute vectors in the training split.

2.3 RIVAL10

2.3.1 Overview

RIVAL10 differs from previous attributed datasets in that it provides *attribute-specific* localizations. That is, for every positive instance of an attribute, a binary segmentation mask identifies the image region in which the attribute occurs.

Perhaps, the most similar dataset in this regard is the recent Fashionpedia [55], a dataset providing attributes and localizations of 27 apparel categories. However, the dataset is proposed for the fashion domain which limits its utility for general purpose object recognition task. To the best of our knowledge, RIVAL10 is the first *general domain* dataset to provide both rich semantic attributes and localization, the combination of which we envision to aid in analyzing the robustness and interpretability of deep networks. While other datasets used for semantic segmentation and object detection go beyond single label annotations [56, 57, 58], they are not designed with classifiers specifically in mind, like RIVAL10.

Classes were chosen to be aligned with CIFAR-10 to enable analyzing the existing architectures and training techniques developed for the object recognition task. In particular, the classes we provide are: *bird, car, cat, deer, dog, equine, frog, plane, ship, truck*. We collected the following attributes for these object categories: *beak, colored-eyes, ears, floppy-ears, hairy, horns, long, long-snout, mane, metallic, patterned, rectangular, tail, tall, text, wet, wheels, wings*. Some attributes were inspired from [31].

We chose attributes to be intuitively informative, capturing semantic concepts that humans may allude to in classifying RIVAL10 objects. While the attributes contain some redundant information, they are nonetheless discriminative in the sense that a linear classifier on attributes achieves 93.3% test accuracy. We visualize attribute correlations and class-wise frequencies in Figure 2.3.

2.3.2 Data Collection

All images were sourced from ImageNet [14]. The images used in each RIVAL10 class were derived from pairs of related ImageNet classes. In other words, 20 classes from Imagenet were used to build the 10 RIVAL10 classes. To collect our attributes and localizations, we hired workers from Amazon Mechanical Turk (AMT). Data collected through AMT without careful control may be of low quality. To encourage quality annotations, we utilize strategies recommended by the HCI community [59]: providing detailed instructions, *screening* workers for aptitude, and monitoring worker performance with attention checks.

Binary attributions were collected first. Workers were required to pass a qualification test of 20 images with known ground truth attributes: only workers who achieved a minimal overall

precision and recall of 0.75 were hired for full data collection. Because the task of segmentation is more involved than indicating whether or not an attribute is present, we required a second qualification test, assessing annotation quality by computing intersection-over-union (IOU) of the submitted attribute masks with ground truth masks. Workers were required to complete five segmentations with an average IOU of at least 0.7.

To ensure that quality is maintained in both the attribution and segmentation phases, roughly 5% of images provided to workers to annotate already had ground truth labels. These so-called attention checks allowed for the monitoring of annotation quality during the collection process. In the first stage of collecting binary attribute labels, the average precision and recall scores were 0.81 and 0.84 respectively. For each positive instance of an attribute marked in the first phase of data collection, an attribute segmentation was collected in the second phase. Completeing attribute segmentations in a second pass allowed for the review of the binary attributions and the removal of any false positives. Average IOU of attention checks completed during the second phase of data collection was 0.745. Further details on our data collection pipeline, including images of instructions shown to workers, payments, and quality-assurance metrics, can be found in the appendix.

2.4 Models

In our analyses, we focus on ResNets and Vision Transformers[16, 21]. We inspect ResNets trained (i) in a standard supervised fashion, (ii) adversarially via ℓ_2 projected gradient descent[60], and (iii) contrastively (i.e. no direct label supervision), with SimCLR and CLIP[12, 13]. We also consider CLIP Vision Transformers, as well as standard Vision Transformers (ViT) and Data ef-

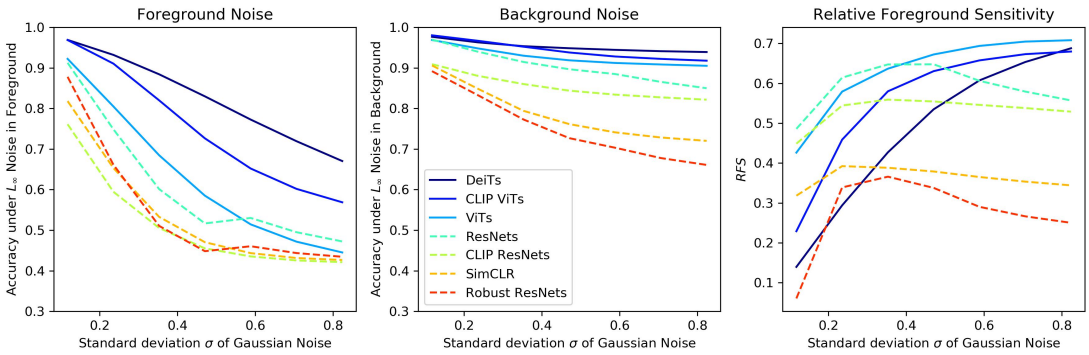


Figure 2.4: Accuracy under noise in foreground (**left**) and background (**middle**) at various noise levels. Models are grouped by architecture and training procedure, with a curve corresponding to the average over all models in a group. (**Right**): RFS by group.

ficient Image Transformers (DeiT)[61]. DeiTs differ from ViTs primarily in their training set, solely using ImageNet-1k while ViTs used ImageNet-21k. To make up for not having the inductive biases of ResNets, ViTs increased the amount of training data, while DeiTs instead rely upon extensive augmentation. All other models, with the exception of those trained with CLIP, use ImageNet-1k as the pretraining set. CLIP, on the other hand, uses a much larger dataset of images and associated text. A full discussion on models is offered in the appendix.

To perform classification on RIVAL10 dataset, we append a linear head to the penultimate layer of each base model. Only the linear head is fine-tuned on RIVAL10's train split (i.e. other weights are frozen), so to preserve the feature space learned in the original pretraining. All models achieve upwards of 90% accuracy on the RIVAL10 test set, essentially controlling for classification ability. We note that while there is leakage between ImageNet-1k and the RIVAL10 test set, the purpose of this study is not to improve model's predictive accuracy directly, but instead to better understand the information used in making predictions.

Recently, a number of works compare the robustness of ViTs to ResNets. While there are mixed findings on adversarial robustness [62, 63], there is agreement that ViTs have stronger out-

of-distribution generalization, likely due to self attention [64, 65]. In contrast, our work focuses on relative robustness to noise in foreground and background regions.

2.5 Foreground and Background Sensitivity

2.5.1 Noise Analysis

We add noise to the foreground and background separately to see how corrupting each region degrades model performance. Consider a sample $\mathbf{x}$ with a binary object mask $\mathbf{m}$ where $\mathbf{m}_{i,j} = 1$ if the pixel $\mathbf{x}_{i,j}$ is a part of the object. We first construct a noise tensor $\mathbf{n}$ that has pixel values drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$, where σ is a parameter controlling the noise level. Then, we obtain noisy-background $\tilde{\mathbf{x}}_{bg}$ and noisy-foreground $\tilde{\mathbf{x}}_{fg}$ samples as:

$$\tilde{\mathbf{x}}_{fg} = \text{clip}(\mathbf{x} + \mathbf{n} \odot \mathbf{m}),\ \tilde{\mathbf{x}}_{bg} = \text{clip}(\mathbf{x} + \mathbf{n} \odot (\mathbf{1} - \mathbf{m}))$$

where $\odot$ is the hadamard product, and 'clip' refers to clipping all pixel values to the $[0, 1]$ range. We add Gaussian noise so to preserve the image content. Note that additive pixel-wise noise leads to the same magnitude of perturbation in the foreground and background under the ℓ_∞ norm. We also repeat our analysis with ℓ_2 normalized noise (presented in the appendix) to avoid a bias against larger regions and obtain similar results.

We seek to quantify the sensitivity of a model to foregrounds relative to its sensitivity to backgrounds. To this end, we introduce *relative foreground sensitivity* (RFS). Let a_{fg} and a_{bg} denote accuracy under noise in the foreground and background, respectively, and $\bar{a} := (a_{fg} + a_{bg})/2$ denote their mean (referred to as the general noise robustness). We then define RFS for a

model $\mathbf{F}$ as

$$RFS(\mathbf{F}) = \frac{a_{bg} - a_{fg}}{2\min(\bar{a}, 1 - \bar{a})}.$$

Essentially, RFS normalizes the gap in model performance under foreground and background noise by the total possible gap, given the general noise robustness of the model. In Figure 2.2, RFS takes on the geometric meaning of the ratio between the distance of (a_{fg}, a_{bg}) to $(\bar{a}, \bar{a})$, to the largest possible distance from the diagonal in the unit square for a point with general noise robustness $\bar{a}$. The scale factor in the denominator gives RFS a range of $[-1, 1]$.

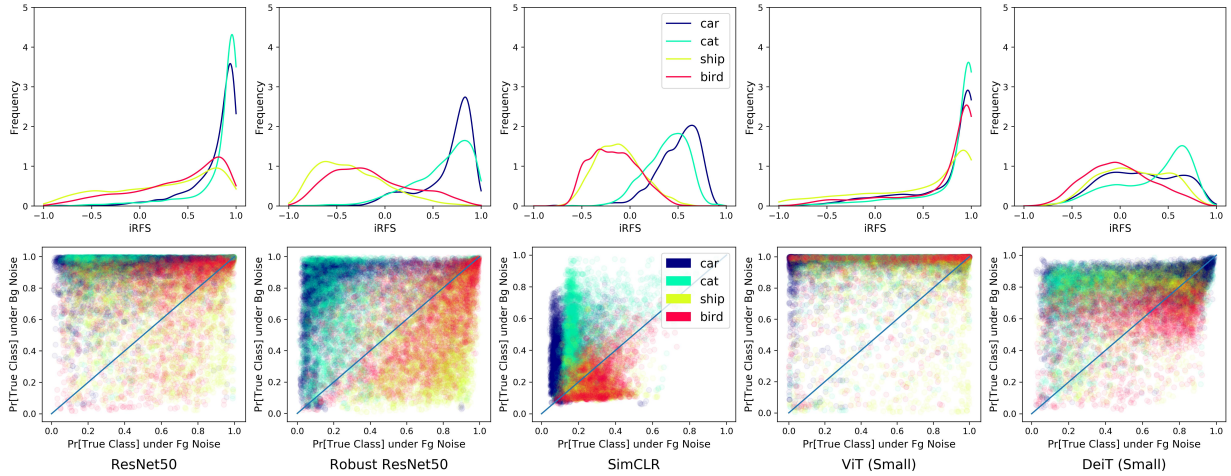


Figure 2.5: Relative foreground sensitivity per instance for four classes and five models of roughly equal size. (**Top**): Histogram of $iRFS$; positive denotes greater foreground sensitivity. (**Bottom**): Scatter; top left indicates high relative foreground sensitivity. Across models, ships and birds have low foreground sensitivity, often being more sensitive to noise in the background than the foreground.

We also consider an instance-wise version, $iRFS$, defined for a model $\mathbf{F}$ *and* a sample $\mathbf{x}$. Here, we use the probability that model $\mathbf{F}$ predicts sample $\mathbf{x}$ to belong to its true class as the measure of model performance instead of accuracy. Let p_{fg} and p_{bg} denote this probability for

$\tilde{\mathbf{x}}_{fg}$ and $\tilde{\mathbf{x}}_{bg}$, respectively. Thus, with $\bar{p} := (p_{fg} + p_{bg})/2$,

$$iRFS(\mathbf{F}, \mathbf{x}) = \frac{p_{bg} - p_{fg}}{2\min(\bar{p}, 1 - \bar{p})}.$$

In our experiments, we consider seven equally spaced noise levels from $\sigma = 30/255$ to $210/255$. For each sample in the test set, we take ten trials of adding noise to the foreground and background separately *per noise level*. RIVAL10's test set consists of roughly $5k$ images, so for each model type, we assess $5k \times 7 \times 10 = 350,000$ trials in total.

2.5.2 Empirical Observations

Fig. 2.2 shows different models have vastly different performance in terms of both general noise robustness and relative foreground sensitivity. In Figure 2.2, transformers generally lie further up the main diagonal than ResNets, corroborating observations that transformers are more robust to common corruptions [66]. Increasing model size improves general robustness, though it does so more for transformers than ResNets. Models lie at different distances orthogonal to the diagonal as well, indicating architecture and training procedure affect relative foreground sensitivity.

In Figure 2.4, we categorize model types based on architecture and training procedure, averaging RFS over groups to reveal general trends. **Robust ResNets have the lowest** RFS, much lower than standard ResNets, a somewhat surprising result given that background reliance has been thought to be linked to increased adversarial vulnerability in the past [39, 67]. SimCLR has the next lowest RFS overall, and generally, **contrastive training procedures (CLIP, SimCLR) seem to reduce** RFS in both ResNets and ViTs.

In comparing transformers to ResNets overall, we see at low noise levels, transformers sometimes have lower RFS than ResNets. Interestingly, **as noise level rises, RFS in transformers increases** as well, while RFS is mostly stable for ResNets. This suggests that transformers can adaptively alter the attention paid to different image regions based on the level of corruption. Comparing between transformers, we see DeiTs with much lower RFS than ViTs, suggesting that the heavy augmentations DeiTs leveraged to achieve increased data efficiency may have also made the models much more sensitive to backgrounds.

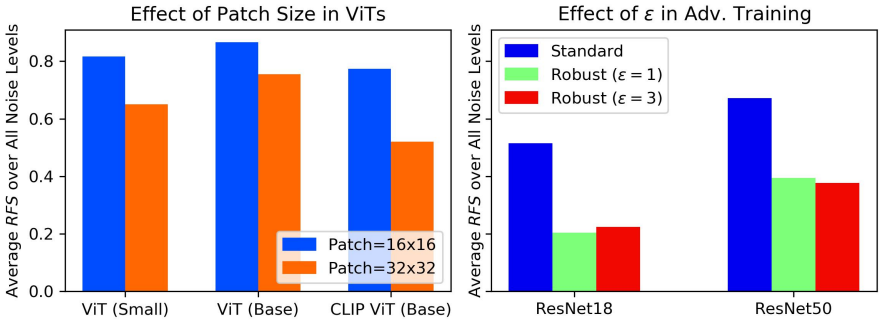


Figure 2.6: Controlled ablation studies. Average RFS over all noise levels presented for brevity. (**Left**): Increasing patch size in ViTs decreases relative foreground sensitivity. (**Right**) Robust models are much less relatively sensitive to foregrounds, but ϵ used in adversarial training does not affect RFS much.

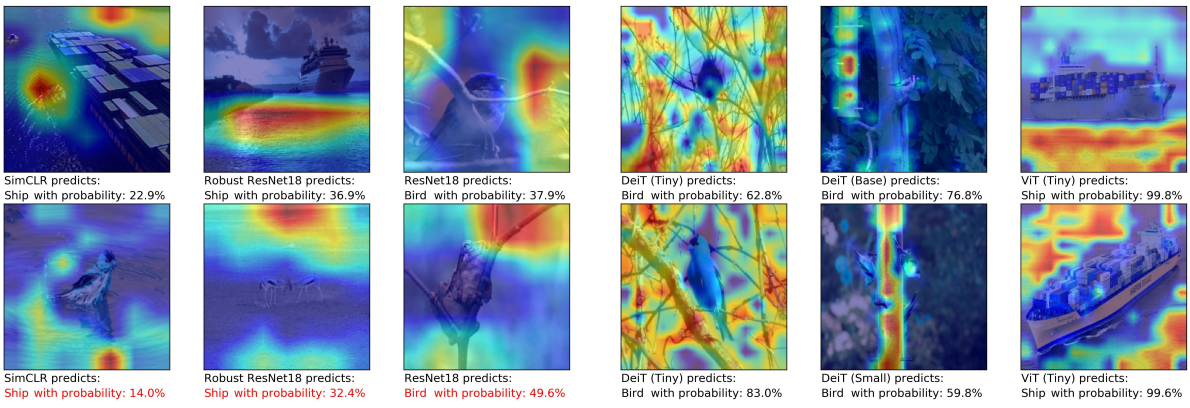


Figure 2.7: Instances of images with low saliency alignment, highlighting spurious features of water for ships, and branches and bird feeders for birds. (**Left**): Spurious features leading to misclassification (in red). (**Right**): Other instances of spurious features.

In Figure 2.6, we more closely inspect the effect of patch sizes in ViTs and the attack

budget ϵ used in adversarial training (which affects accuracy-robustness trade-off). We find that increasing the patch size in ViTs from 16×16 to 32×32 reduces RFS when averaged over all noise levels. The robustness ablation affirms that robust ResNets are much less relatively sensitive to foregrounds than standard ResNets, though the attack size seen in training does not seem to significantly affect RFS.

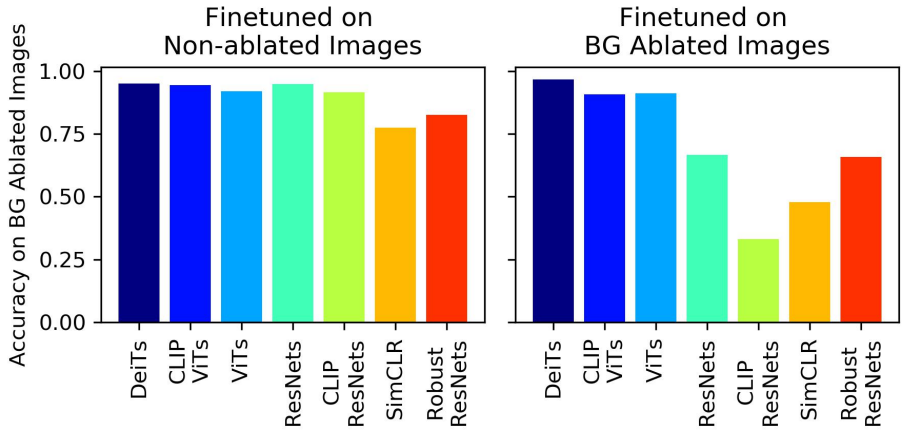


Figure 2.8: Model accuracy on images with backgrounds ablated via graying. The right plot shows accuracies for models finetuned on the images with backgrounds ablated. Only transformers can fit a linear layer on the features of background ablated images without compromising performance.

Moving away from comparing models, in Figure 2.5, we see **foreground sensitivity is largely affected by class**. In particular, across models of roughly equal size, ships and cats are often more sensitive to background noise, suggesting models learn to utilize background content more than foreground content in recognizing them. The class distinction is less pronounced in DeiTs and ViTs, with ViTs assigning high foreground sensitivity for all classes, and DeiTS having mixed sensitivity across classes, with many negative $iRFS$ scores (i.e. higher background sensitivity).

2.5.3 Removing Backgrounds Entirely

We also inspect the accuracy of models on images with backgrounds grayed out, similar to [39], though now considering ViTs, CLIP, and SimCLR, which had not been developed at the time of their study. Also, the rich annotations of RIVAL10 allow for going beyond foreground or background ablation (see the appendix for a discussion of attribute removal). Ablation via graying can be thought of as another kind of noise, where all pixels are smoothed to 0.5. In Figure 2.8, the left plot reveals that Robust ResNets and SimCLR see the largest drops in accuracy when evaluated on images with grayed backgrounds. Transformers do well on ablated images, consistent with the observation that transformers had high RFS at the largest noise levels. Furthermore, when we attempt to fit a linear layer to classify background-ablated images, only the features from transformer models are sufficiently informative to have high linear classification accuracy. Thus, while transformers make use of backgrounds, they still retain significant foreground information in their feature space. This result suggests transformers are much more robust to localized distribution shift. That is, distribution shift in one region (the background) may affect model perception of other unperturbed regions much less in transformers than ResNets.

2.5.4 Saliency Alignment

To complement the noise analysis, we use GradCAM [22] to assess the amount of saliency that models place on foreground pixels. RIVAL10's object segmentations allow us to automatically quantify saliency alignment with foregrounds, removing the need for human inspections. Sorting samples based on saliency alignment reveals failure modes, where models deem background regions as highly salient. We present several metrics to assess saliency alignment in the

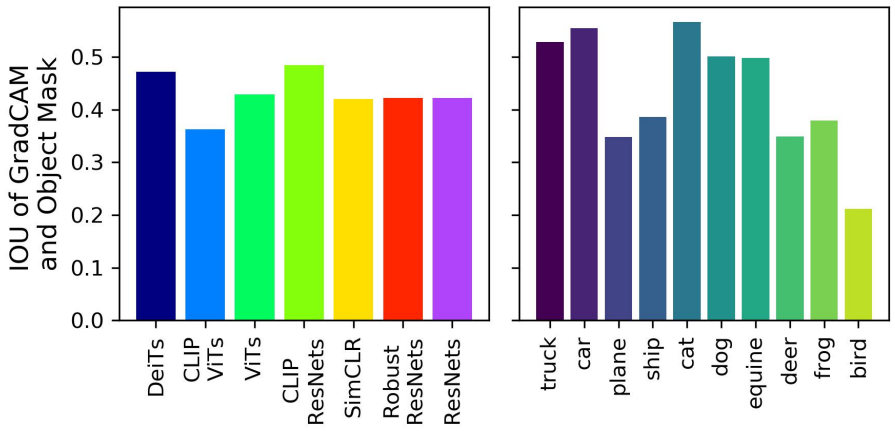


Figure 2.9: Alignment of binarized saliency maps with object segmentation masks, measured by intersection over union (IOU). Averaged over models (**left**) and object classes (**right**).

appendix. We find that extracting samples with the lowest difference in average pixel saliency in foreground and background yield the most interesting failure modes. We present examples selected this way in Figure 2.7, highlighting spurious background features that contribute to the low RFS of ships and birds observed across models in Figure 2.5. Specifically, models look for water and coasts when classifying ships, and twigs and branches when classifying birds.

In Figure 2.9, we appeal to the standard metric intersection-over-union (IOU). Saliency maps are binarized using a threshold of 0.5 before being compared with object segmentation masks. On average, saliency alignment is similar across models, despite their being large differences in RFS identified in the noise analysis, suggesting saliency maps may give an incomplete picture of model sensitivity. In comparing saliency alignment across classes, we see much larger differences, emphasizing the result that **class matters** when it comes to background reliance.

2.6 Conclusion

We present **RI**ch **V**isual **A**ttributes with **L**ocalization (RIVAL10), and quantitatively assess sensitivities of state-of-the-art models under noise corruption. Specifically, we find adversarially or contrastively training ResNets leads to reduced relative foreground sensitivity. Further, we observe that transformers adaptively raise foreground sensitivity as noise level increases, while ResNets do not. By applying automated alignment metrics to saliency maps, we reveal instances of spurious background features used by models. Lastly, we observe promising evidence that neural node attributions based on top activating images generalize to instances unseen during attribution. We hope RIVAL10's rich annotations lead future studies to gain new quantifiable insights on the behavior of deep image classifiers.

Chapter 3: Hard ImageNet: Segmentations for Objects with Strong Spurious Cues

This chapter details a NeurIPS '22 paper co-authored by Sahil Singla and Soheil Feizi [68].

Deep classifiers are known to rely on spurious features, leading to reduced generalization. The severity of this problem varies significantly by class. We identify 15 classes in ImageNet with very strong spurious cues, and collect segmentation masks for these challenging objects to form *Hard ImageNet*. Leveraging noise, saliency, and ablation based metrics, we demonstrate that models rely on spurious features in Hard ImageNet far more than in RIVAL10, an ImageNet analog to CIFAR10. We observe Hard ImageNet objects are less centered and occupy much less space in their images than RIVAL10 objects, leading to greater spurious feature reliance. Further, we use robust neural features to automatically rank our images based on the degree of spurious cues present. Comparing images with high and low rankings within a class naturally reveals the exact spurious features models rely upon, and shows classifiers suffer reduced accuracy when spurious features are absent. With Hard ImageNet's annotations and evaluation suite, the community can begin to address the problem of learning to detect challenging objects *for the right reasons*, despite the presence of strong spurious cues.



Figure 3.1: *Hard ImageNet* classes. Object segmentation masks are overlaid in purple.

3.1 Introduction

Deep learning based image classifiers are effective but brittle to distribution shift, leading to serious issues when models are deployed to sensitive applications like medicine [69, 70]. The reliance of models on spurious features, which are predictive of class labels in training data but are irrelevant to the true labeling function, contributes to reduced distributional robustness. The lack of interpretability of deep models is a bottleneck to understanding causes and degree to which models rely on spurious features, especially on real data. Some datasets have been proposed where spurious features are manually annotated or injected into synthetic [71, 72] or application-specific [33, 73] datasets, though insights from these benchmarks may have limited transferability to more general domains.

Recently, [1] apply a mostly automated procedure for neural network interpretability [48] to discover and annotate core (i.e. essential to the class label) and spurious features at scale. [74] expand this analysis to softly segment core input regions in nearly all of the ubiquitous benchmark dataset, ImageNet [14]. Their method leverages the activation maps of neural features that highly contribute to the activation of a given class logit and detect core features for that class. However, for 15 classes, all of the inspected features were spurious, making the core-feature-based segmentation scheme infeasible, while suggesting the presence of uniquely strong spurious cues within this subset of ImageNet.

In this paper, we first gather object segmentations for these 15 classes, forming the *Hard ImageNet* dataset. We then curate an evaluation suite of ablation, noise, and saliency based analyses that leverage object segmentations to assess the degree to which models rely on spurious features. With this benchmark, we show that models rely on spurious features much more for Hard ImageNet objects than more typical objects, as represented by the RIVAL10 dataset [15], who's classes appear in multiple standard benchmark datasets (ImageNet, CIFAR10 and STL10 [26, 75]). We then investigate properties of Hard ImageNet objects, observing that they are significantly smaller and less centered, likely leading to models favoring spurious features. Further, we complement our instance-wise object segmentations by ranking each image within a class based on the strength of spurious cues present, leading to clearer visualizations of the specific spurious features involved. The image rankings also open the door to last-layer retraining of models on data where the presence and absence of spurious features is balanced, an efficient new technique for reducing spurious feature reliance [76].

With Hard ImageNet, we go beyond single-label annotations to shed new insights on the data and model conditions that lead to increased spurious feature reliance in deep image clas-

sifiers. We hope our evaluation suite and challenging dataset can inspire new and improved training procedures that result in models that classify *for the right reasons*, towards more reliable and trustworthy AI.

3.2 Related work

The tendency of models to rely on spurious features due to simplicity bias [77] or short-cut learning [5] has garnered much attention, due to its consequences on fairness and generalization, leading to the proposal of many algorithms to mitigate spurious dependencies [72, 78, 79, 80, 81, 82]. However, in comprehensive evaluations, the gains of these methods over empirical risk minimization are often marginal or inconsistent [83, 84]. Recently, a simpler approach for reducing spurious feature reliance has been proposed, where only a final linear layer is retrained over data that balances the presence of spurious cues, thus encouraging models to downweight spurious features, which are less predictive of class labels in the balanced set [76]. A bottleneck to this approach, along with many of the other mentioned algorithms, is that they require knowledge of the degree to which spurious cues are present in each sample; the image rankings of our dataset may address this need.

Several datasets have been created to aid in studying spurious feature use, including many with synthetically-injected spurious correlations [71, 72, 85]. A noteworthy example is ImageNet-9, which swaps backgrounds across classes to assess background sensitivity, using analyses similar to our ablation studies [39]. While insightful, the use of simple classification tasks and unnatural data in these synthetic sets may limit their impact in practice. A number of real-world datasets have also been proposed [33, 73, 86, 87], though they are usually very specific to the

domain of their application.

Recent work appeals to additional annotations to assess model sensitivity to various input regions via noise-based analyses [15, 48]. Notably, [74] annotate core and spurious input regions for nearly all of ImageNet in a mostly automated procedure, leveraging neural feature visualizations [48] and the improved interpretability of an adversarially robust network [88], resulting in *Salient ImageNet-1M*, a large-scale, natural, and general dataset for evaluating and improving model reliance on spurious features. Namely, they used activation maps of robust neural features corresponding to core features as soft segmentations of core regions. However, for 15 classes, all neural features inspected were spurious, making their segmentation framework inapplicable. Arguably, these classes are the most interesting, as the underlying robust network used for segmentation relied on spurious features the most for these classes, indicating that this subset of data contains the strongest spurious cues. We manually collect object segmentations for these 15 classes, design a suite of evaluation metrics, and study the properties of this data that leads to heightened spurious feature reliance.

3.3 Hard ImageNet Dataset

3.3.1 Overview

Our dataset, *Hard ImageNet*, consists of images from 15 ImageNet [14] synsets. These classes are *Dog Sled, Howler Monkey, Seat Belt, Ski, Sunglasses, Swimming Cap, Balance Beam, Gymnastic Horizontal Bar, Patio, Hockey Puck, Miniskirt, Keyboard Space Bar, Volleyball, Baseball Player,* and *Snorkel*, selected because for each class, all features annotated in [74] were spurious with respect to the class. Specifically, [74] inspect the five neural features (i.e. nodes in

the penultimate layer) of an ℓ_2 adversarially trained ResNet50 per class that contribute most to the activation of the corresponding logit. Each class-feature pair is annotated as *core* or *spurious*, based on if the detected feature is essential to the class label. Over all classes, the vast majority of class-feature pairs inspected were deemed *core*. Thus, having all five most important features serving *spurious* roles suggests that each Hard ImageNet class uniquely has very strong spurious cues that models favor over core features.

We collect object segmentation masks for these 15 classes, resulting in a dataset with $19,097$ training samples and 750 validation samples. We maintain the same train/validation split as ImageNet. Figure 3.1 visualizes an example and its object segmentation for each Hard ImageNet class. Additionally, we provide class-wise rankings for each image in the Hard ImageNet training set based on the strength of spurious cues present in the image. We base these rankings on the activations of the annotated neural features for each class (see Section 3.6 for more details).

Impact. First, Hard ImageNet segmentations complete the *Salient ImageNet* dataset, which allows for the exploration of going beyond single-label supervision in training ImageNet models to predict *for the right reasons*. That is, with object segmentations available during training, models can be guided to rely more on core features than spurious ones, leading to improved generalization to domains where spurious correlations are broken. Moreover, Hard ImageNet annotations shed insight on the conditions for models and data under which spurious features are favored: through our suite of evaluation metrics (see Section 3.4), the effect of training procedures and model architectures on spurious feature reliance can be compared, while analysis of segmentation masks reveal how the shape, size, and location of objects in images affects model sensitivity to the object (see Section 3.5).

3.3.2 Collection Procedure

We collect Hard ImageNet object segmentations over Amazon Mechanical Turk. To ensure quality, annotations are collected in many phases. First, workers must pass a *qualification exam*, consisting of segmenting one sample from each class. All workers take the same qualification exam, and only workers who achieve an intersection-over-union (IoU) score of at least 0.65 pass. For reference, any IoU greater than 0.5 is considered successful in the Pascal VOC object detection evaluation [89]. Workers meeting this bar demonstrate proficiency with the Mechanical Turk platform and sufficient understanding of the relevant objects to be segmented. Then, workers sign a *consent form* explaining the nature of and purpose for their work. We also share answers to the qualification exam and tips for improved segmentations. Finally, qualified and consenting workers move on to the full data collection stage, where we release batches in rounds, monitoring quality with *attention checks* (randomly inserted images with ground truth segmentations) and updating class-specific tips based on errors observed on attention checks. We place attention checks with a frequency of 10% for the validation split and 5% for the training split. The average IoU score on attention checks was 0.76. Workers with very low quality on attention checks were removed from the study, and their segmentations were redone in later rounds. Also, each image in the validation set is annotated five times, with annotations consolidated via a pixel-wise majority vote to correct for errors in any individual segmentation. Figure 3.2 shows the IoU scores of each validation set segmentation to the

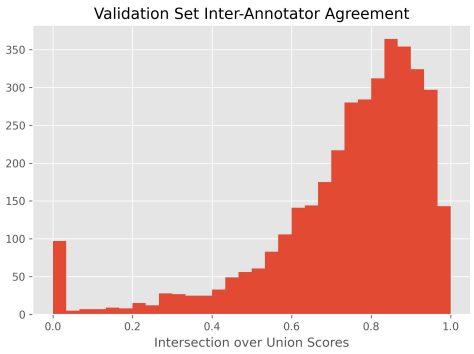


Figure 3.2: Agreement of individual segmentations with final averaged segmentation for Hard ImageNet validation split.

consolidated final validation masks. The vast majority of segmentations are highly agreed upon, with a small minority having near zero agreement, indicating that the extra rounds of annotation had a meaningful corrective impact.

Throughout our data collection, we strive to create a transparent and collaborative working environment. In the consent/information stage, we invite feedback and explicitly encourage our workers to directly contact us with questions. Moreover, we reward workers with fair salaries (estimated to be $\$12$ to $\$16$ an hour depending on annotation speed) and bonuses (between 5% and 7% depending on number of annotations) for continued work. We also invite workers who previously partook in RIVAL10 [15] annotation, as their experience from a similar task leads to improved annotation quality, and rewarding loyalty improves worker well being.

3.4 Benchmarking Spurious Feature Reliance

We compile a suite of evaluation metrics that use object segmentations for assessing image classifier reliance on spurious features. We focus this section on demonstrating the significantly higher degree of spurious feature reliance for models on Hard ImageNet compared to other more typical data. As a baseline, we use RIVAL10 [15], which is comprised of images from 20 ImageNet synsets, selected so that each synset corresponds to a class from CIFAR10 [26] (i.e. two synsets per CIFAR10 class). Importantly, RIVAL10 includes an object segmentation per image, collected in a similar fashion to Hard ImageNet's annotations. We consider RIVAL10 data to be more typical of what is commonly studied in image classification, as many or all of the classes also appear in the standard benchmark datasets CIFAR10 and STL10. We note that we use samples in RIVAL10 that were originally in ImageNet's validation split, as opposed to using

RIVAL10's own train/test split, which leaks ImageNet training samples to their test split. For Hard ImageNet, we also evaluate over images originally in ImageNet's validation split, which corresponds to Hard ImageNet's validation split.

Across our evaluations, we use a convolutional neural network (ResNet50 [16]) and a vision transformer (small DeiT [61]) of roughly equal size, both pretrained on ImageNet. To evaluate these models 'off the shelf', we map samples from Hard ImageNet and RIVAL10 to their original ImageNet class indices. Thus, for RIVAL10 evaluation off the shelf, we actually consider 20 target classes, and so we refer to this as RIVAL20. We also include finetuned versions where a final linear layer is trained for each model atop representations from the corresponding fixed feature encoder. For RIVAL10, we finetune for both 10-way and 20-way classification. While the emphasis of this section is *data*, in practice our evaluation metrics are intended for comparison of *models* of diverse types.

3.4.1 Ablation

Object segmentation masks allow for removal of the object corresponding to the class label for any sample. Classifying an ablated image to its original class indicates that the model uses background information as it does not require the object's presence to predict its class. While these ablated images are out of distribution, accuracy drop due to ablation can still inform the degree to which the model relies on spurious background features. We consider three types of ablation. First, we replace all object pixels with 0.5, graying out the object. Second, we replace all pixels in the *bounding box* of the object with gray, thus also removing object shape information. Finally, we replace the bounding box of the object with a tile adjacent to the object in the image.

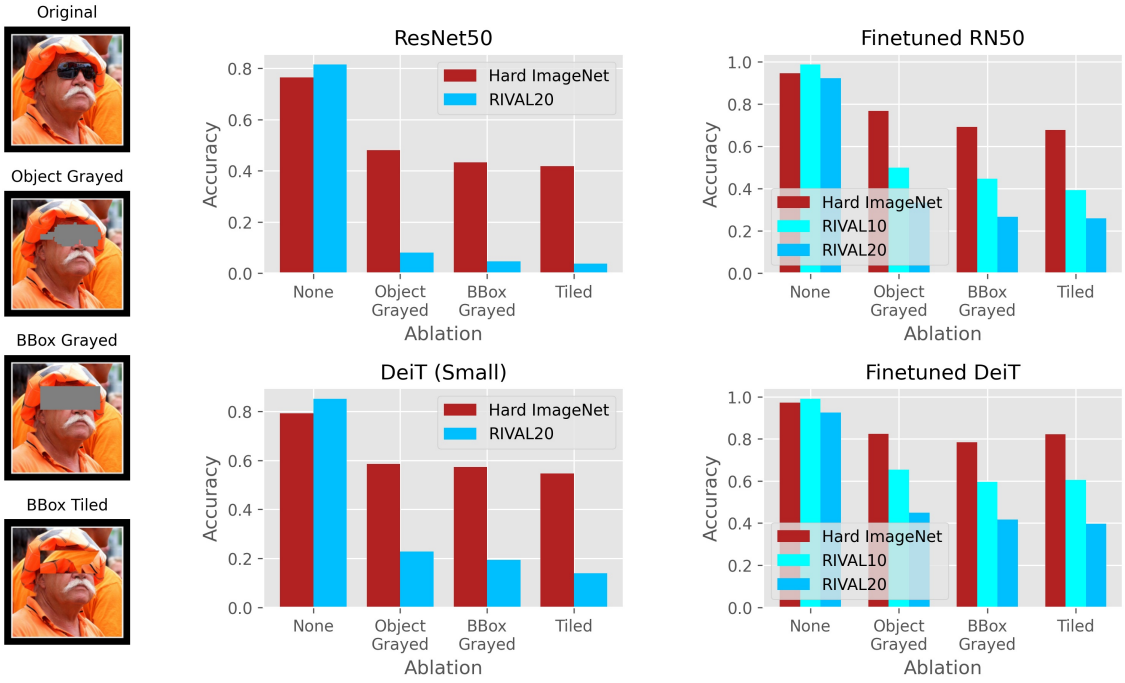


Figure 3.3: Accuracy under ablation. Accuracy drops much less when Hard ImageNet objects are ablated than when RIVAL10 objects are ablated. Example ablations for class *sunglasses* shown left.

In cases where a tile is smaller than the bounding box, we repeat the tile to fill the region, as in [39]. Note that all ablations replace each instance of an object in an image separately (i.e. we consider multiple boxes bounding each instance as opposed to one box containing all instances) so to retain as much background as possible.

Figure 3.3 shows that ablation leads to a much smaller accuracy drop for Hard ImageNet objects than for RIVAL10 objects, in both the finetuned and off the shelf case, and for all types of ablation. In fact, for an off the shelf DeiT (i.e. performing 1000-way classification), Hard ImageNet images are predicted to their class over 50% of the time under all ablations, despite the fact that the object essential to the class label is actually missing, rendering the predictions inaccurate. This is several times larger than the rate for RIVAL20. In finetuned models, accuracy under ablation is still significantly larger for Hard ImageNet, though this gap is smaller.

We note that random guessing yields much higher accuracy in these cases because the number of classes is multiple orders of magnitude smaller. Because number of classes varies across the three datasets considered, direct comparison is more challenging, but Hard ImageNet accuracy is still substantially larger than RIVAL10 ablated accuracy, where random guessing achieves 5% higher accuracy than on Hard ImageNet.

3.4.2 Relative Foreground Sensitivity

We now turn to the noise-based metric *relative foreground sensitivity* (RFS), introduced with RIVAL10 [15]. To compute RFS, equal amounts of Gaussian noise are added to foreground and background regions. Then, the gap in accuracy drops (i.e. how much more accuracy drops due to foreground noise than background noise) is normalized to allow for comparison across models with varying noise robustness, as derived extensively in [15] and [74]. Higher RFS scores entail greater sensitivity to noise in foregrounds relative to backgrounds. We consider adding both ℓ_∞ and ℓ_2 noise, where in the latter case, we fix the ℓ_2 norm of noise added in both backgrounds and foregrounds. These types of noise have opposite size biases, as ℓ_2 noise results in a higher ℓ_∞ norm of noise is smaller regions, while ℓ_∞ noise results in a higher ℓ_2 norm of noise in larger regions.

In figure 3.4, we see that for all models and across noise levels, RFS is substantially lower when evaluated on Hard ImageNet than on RIVAL10 or RIVAL20. In fact, RFS **is negative in many cases**, indicating that noise in the background reduces accuracy more than noise in the foreground. For finetuned models at low levels of ℓ_2 normalized noise, RFS is comparable across datasets, but otherwise, models appear to be significantly more sensitive to noise in backgrounds

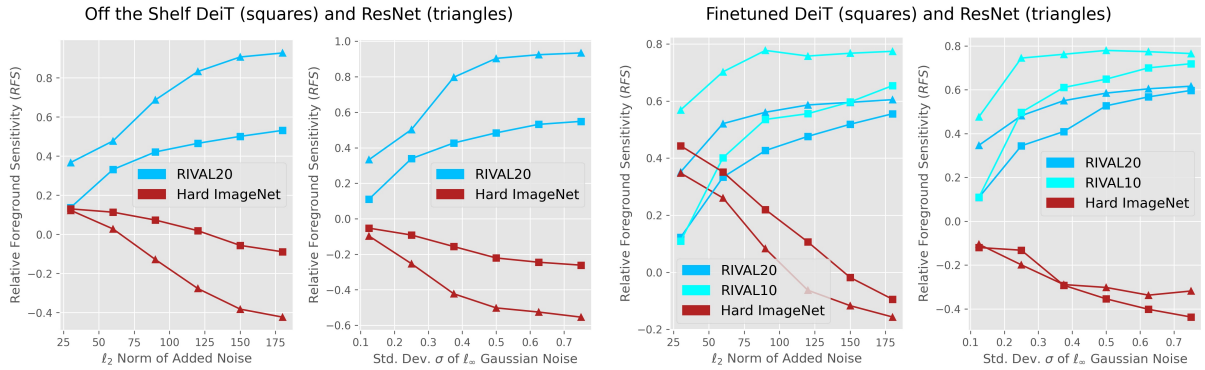


Figure 3.4: Relative Foreground Sensitivity (RFS) for DeiT and ResNet, off the shelf (left) and finetuned (right). RFS for Hard ImageNet is significantly lower, and often negative, indicating greater sensitivity to background noise than foreground noise.



Figure 3.5: GradCAMs of Hard ImageNet instances with poor saliency alignment for finetuned ResNet50 (left three) and finetuned DeiT (right two). Spurious features of *gymnast, dogs, hockey stick and ice, legs* and *keys* for each class respectively are highlighted.

than they are to the actual object in Hard ImageNet, suggesting that models use object information far less than surrounding context. Arguably, despite achieving high accuracy, this result suggests that models are not actually learning to detect Hard ImageNet objects, instead relying on spurious background cues.

3.4.3 Saliency Alignment

Our third metric makes use of the interpretability method GradCAM [22], which assigns a saliency score to each input pixel, proxying its importance to the prediction. While GradCAMs are qualitative, we perform a quantitative analysis by computing the alignment of GradCAMs to

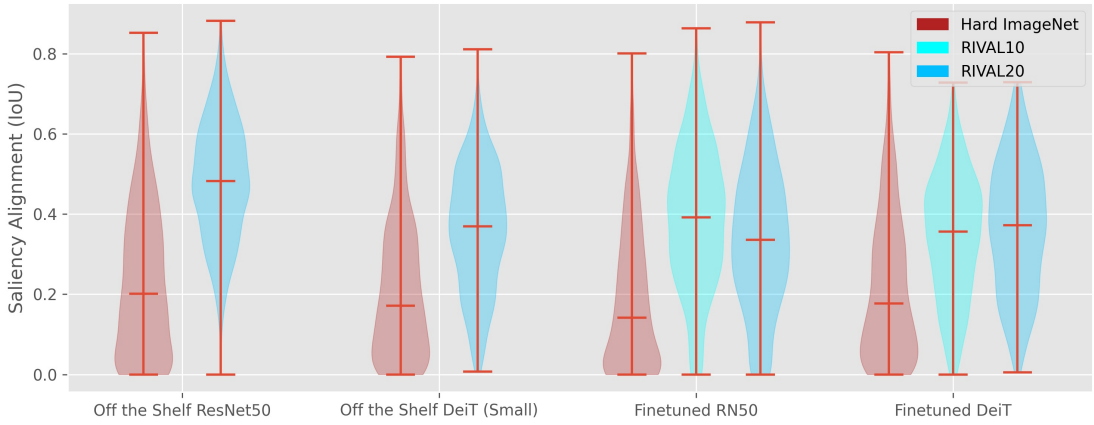


Figure 3.6: Alignment of GradCAM saliencies to object masks. Low alignment indicates more saliency is placed in the background. Across all models, saliency alignment is lowest for Hard ImageNet.

objects. Specifically, we compute IoU of object segmentations and GradCAMs (binarized with a threshold of 0.5) as in [15]. Figure 3.6 shows IoU scores across models are significantly lower for Hard ImageNet objects than RIVAL10 objects, indicating that backgrounds are more salient to Hard ImageNet classification.

With saliency alignment, we can also automatically extract instances with low IoU scores to reveal specific spurious background cues. Some examples are visualized in figure 3.5. Here, we see that other features present in the image but non-essential to the class label are arguably more prominent and easier to recognize than the actual object. For example, the keys on a keyboard are more noticeable than the spacebar below, as they take up more of the image and contain more detail than the uniformly colored spacebar. Further, the location of the objects in these examples makes them appear as though they are in the background, as they reside along the edges of the image, as opposed to the center. We investigate the size and location of Hard ImageNet objects in more detail in the following section.

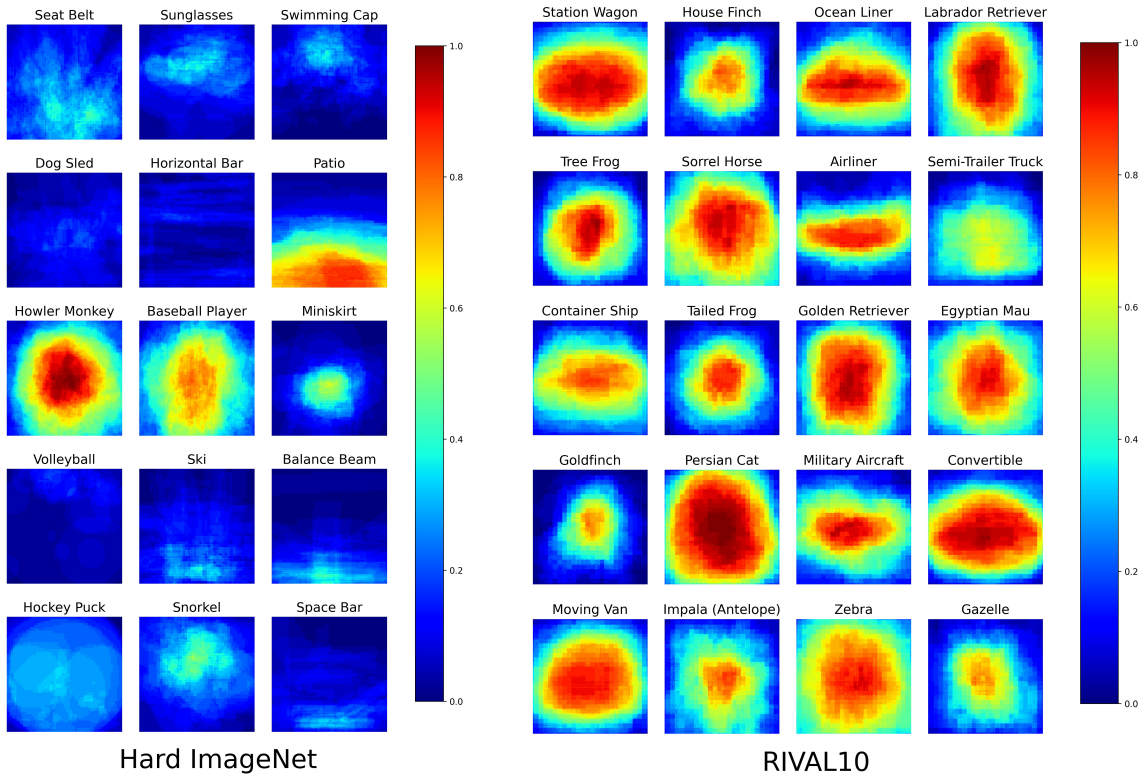


Figure 3.7: Average object position for Hard ImageNet and RIVAL10. Hard ImageNet objects occupy less space and are often not centered, unlike RIVAL10 (and most other ImageNet) objects.

3.5 Hard ImageNet Objects are Smaller and Less Centered

Figure 3.8 shows that compared to RIVAL10, Hard ImageNet objects occupy substantially less space both in their original images and after the standard ImageNet test time augmentation (consisting of resizing and center cropping). Some reasons for this are that Hard ImageNet objects are i. simply smaller (*hockey puck, swimming cap*), ii. oddly shaped so that square crops include lots of background (*spacebar, ski*), iii. often co-occurring with much larger spurious features (*snorkel, balance beam*).

Interestingly, the standard test time augmentation only slightly increases the percent of the image the object occupies, unlike for RIVAL10 objects, which see a larger increase. The lack of

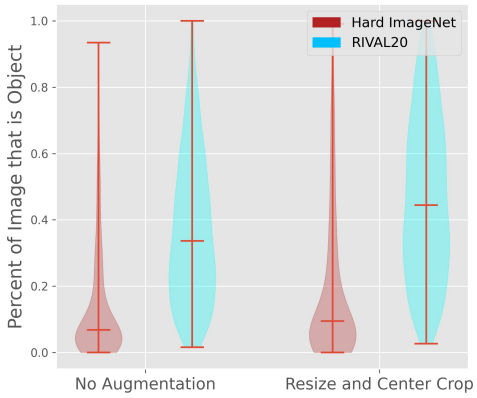


Figure 3.8: Hard ImageNet objects occupy less space in images than RIVAL10 objects both before and after standard augmentation.

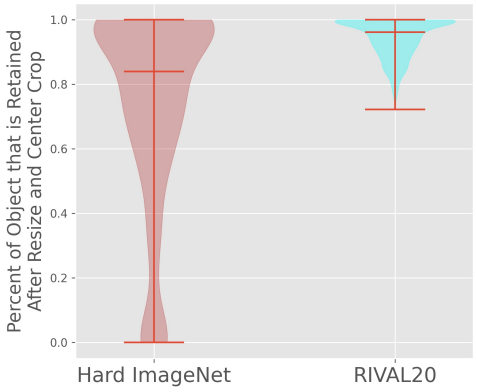


Figure 3.9: Hard ImageNet objects are often substantially cropped out by the standard ImageNet test time augmentation.

a center bias in object locations likely contributes to the reduced benefit of augmentation, as a center crop would not amplify non-centered objects. In fact, we find that this augmentation often removes large portions of Hard ImageNet objects, as shown in Figure 3.9. While in the worst case at most 25% of a RIVAL10 object is lost due to the standard augmentation, much more of the object is lost in Hard ImageNet, including cases where **standard augmentation completely removes Hard ImageNet objects**.

We visualize the class-wise average of object segmentations after standard augmentation in figure 3.7. Indeed, Hard ImageNet objects are smaller and less centered than RIVAL10 objects. For *howler monkey* and *baseball player*, which are relatively larger and more centered, we conjecture that the increased spurious feature reliance is due to easily recognizable spurious cues. Namely, howler monkeys are usually photographed in dense foliage with many leaves and branches. For baseball players, the uniform green grass and brown dirt of the baseball field is likely very easy for a deep network to detect. While patios are not centered, they do take up lots

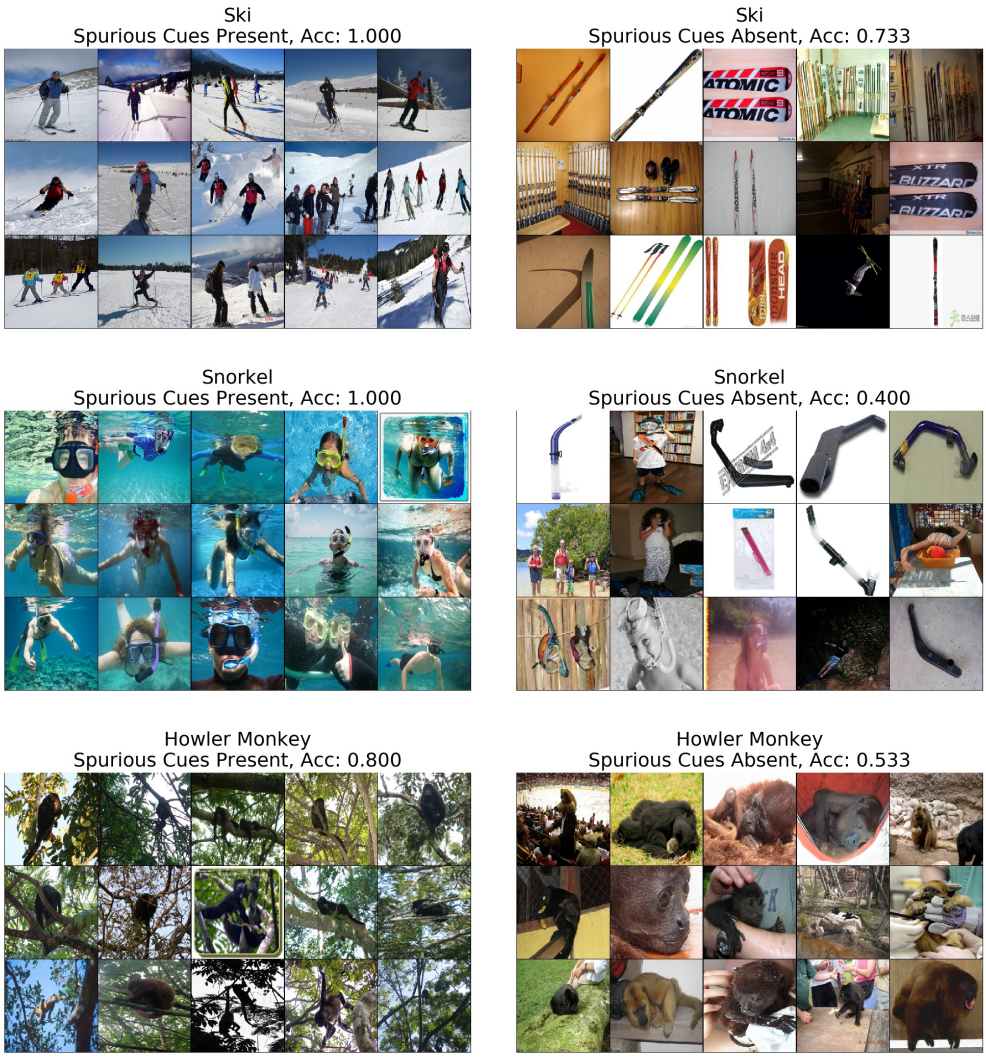


Figure 3.10: Examples of high (left) and low (right) spurious feature presence, based on Hard ImageNet rankings. Accuracies shown are for a standardly trained ResNet50.

of space, but they often co-occur with patio furniture, which models may rely on more so than the patio itself. Finally, we note the smaller size of the finches in RIVAL10; RIVAL10's original analysis observed greater spurious feature reliance in the bird class, further suggesting that small object size leads to greater model use of backgrounds.

3.6 Ranking Images by Spurious Feature Presence

To complement Hard ImageNet segmentation masks, we annotate each image in the training set with a class-wise rank corresponding to the strength of spurious features present in the image. We determine these ranks by appealing to neural features (nodes in the penultimate layer) of the ℓ_2 adversarially trained ResNet50 used to construct the Salient ImageNet dataset [1, 74]. Per class, we first rank all images by feature value on each of the five annotated features (which are all spurious for Hard ImageNet classes). We then sort the average of these ranks to obtain our final image rankings.

Figure 3.10 shows the highest and lowest ranked images for classes *ski, snorkel*, and *howler monkey*. Samples with low spurious rank are classified with significantly lower accuracy. Also, viewing the high and low ranked images reveals the strong spurious cues present in Hard ImageNet. For example, models appear to rely on *snow* and *skiers* to classify *skis*. Similarly, the absence of *people* and *water* for *snorkels* and *trees* for *howler monkey* reduces spurious ranking and classification accuracy. Viewpoint may also be a spurious feature for *howler monkey*, as highly ranked images are all taken from below, with the sky and trees as the backdrop. Viewpoint, along with backgrounds, was observed to be a spurious correlation that contributes to large accuracy drop when broken in ObjectNet [37].

While image ranking facilitates the interpretation of spurious features naturally (i.e. without using visualizations that make artificial changes to the image, which were observed to be less effective in [90]), they can also be used to reduce spurious feature reliance. Namely, retraining a linear layer atop fixed encoders on data that balances the presence and absence of spurious features has been found to greatly improve performance on instances where spurious correlations

are broken [76]. Spurious image ranking allows for the construction of these balanced datasets, explored in the following section.

3.7 Improving Models with Hard ImageNet Annotations

We now explore baseline methods to harness Hard ImageNet's annotations for improved model classification (i.e. with reduced reliance on spurious features). We focus our study on finetuned models pretrained on ImageNet, using ResNet50 and DeiT (Small) as in Section 3.4. We keep features fixed during finetuning, only optimizing the parameters of a new final layer for the 15-way Hard ImageNet classification.

We employ two approaches for mitigating spurious feature reliance. [74] propose **Core Risk Minimization** (CoRM) as an alternative to ERM when segmentations of core (i.e. not spurious) regions are available; Hard ImageNet's object segmentations fulfill this prerequisite. Specifically, the objective of CoRM is to minimize classification error over the distribution of images *with noise applied to non-core regions*, so that the optimal classifier predicts correctly even when spurious features are corrupted. In that work, *random noising*, where small amounts of Gaussian noise are added to non-core regions with probability $p = 0.5$, and *saliency regularization*, where the ℓ_2 norm of the gradient on non-core pixels is added to the classification loss, were applied in tandem to improve relative core sensitivity (an analagous metric to RFS). [76] propose **Deep Feature Reweighting** (DFR), in which retraining a final linear layer using a *balanced dataset* reduces spurious feature reliance. The balanced dataset consists of a subset of the training data containing an equal portion of samples with and samples without spurious features, essentially breaking spurious correlations that impede generalization to minority groups. Using

Hard ImageNet's image rankings, we extract the top and bottom 100 images for each class to form the spurious-balanced subset.

Method		Ablation Accuracies (↓)				RFS (↑)		Saliency (↑)
CoRM	DFR	None (↑)	Gray	Gray BBox	Tile	$\sigma = 0.25$	$\sigma = 0.5$	IoU
Finetuned DeiT (Small)								
✗	✗	**96.79**	84.22	80.48	81.15	−0.19	−0.35	20.90
✓	✗	96.39	**81.02**	78.74	80.75	**0.02**	**−0.19**	21.57
✗	✓	96.66	81.28	**77.01**	77.94	−0.20	−0.33	21.63
✓	✓	96.52	82.35	**77.01**	**77.81**	−0.10	−0.29	**21.99**
Finetuned ResNet50								
✗	✗	94.25	75.94	69.39	67.38	−0.18	**−0.27**	18.44
✓	✗	92.91	76.20	69.12	68.32	**−0.08**	**−0.27**	**20.43**
✗	✓	**94.39**	73.53	67.51	66.71	−0.27	−0.35	18.39
✓	✓	91.31	**72.59**	**63.64**	**63.90**	−0.23	−0.31	20.35

Table 3.1: Final layer retraining improves faithful learning on Hard ImageNet. Results shown for entire benchmark under two different training approaches: i) Core Risk Minimization (CoRM) via *random background noising* and *saliency regularization*, and ii) deep feature reweighting (DFR) using a *spurious-balanced training subset*. We also report results for the combination of the two approaches and ordinary finetuning (as a baseline) under two architectures. Relative Foreground Sensitivity (RFS) is evaluated under two ℓ_∞ noise levels, indicated by σ. Saliency refers to *saliency alignment* as measured by intersection over union (IoU).

Table 3.1 shows that these two methods can considerably reduce model reliance on spurious features, improving numbers across all metrics in our benchmark. Between the two approaches, CoRM appears to lead to more improvement in saliency alignment and RFS, while DFR yields beter results for ablation. Combining CoRM and DFR leads to even better performance with respect to accuracies under ablation. While improvements are at times small, we note that in these experiments, the vast majority of model parameters are left unchanged, as we only train a new final layer. We leave the door open to new approaches for improving the *faithful* learning of

Hard ImageNet objects, including training models from scratch.

3.8 Conclusion

Our benchmark brings a new perspective to classification, as we not only seek models that predict accurately, but also predict for the right reasons. Assessing model performance using accuracy alone can obscure key misconceptions held by models, which may only become apparent when models are deployed to new domains at test time. Moreover, design decisions such as training strategy and architecture may affect the degree to which spurious features are relied upon, as observed in [15]; this dataset and accompanying benchmark can reveal these model differences. Finally, we emphasize the need to understand model behavior under "bad" data; that is, images where the object of interest is not centered or large, unlike most cases. With models becoming increasingly data hungry, it is inevitable that some portion of the data will not capture objects in ideal conditions. Further, certain objects simply are not well suited to be captured prominently (i.e. large and centered) in square photos. Figuring out how to learn to recognize objects from these suboptimal data conditions will be an important challenge to extend the impressive performance of deep classifiers from standard datasets to many more realistic settings. We hope Hard ImageNet can lead to new perspectives on both training and evaluation paradigms for image classification.

Chapter 4: Explicit Tradeoffs between Adversarial and Natural Distributional Robustness

This chapter details a NeurIPS '22 paper co-authored by Kiarash Banihashem and Soheil Feizi [91]. We now expand our analysis of spurious correlations (and the natural distribution shifts that break them) to concurrently investigate adversarial distribution shifts.

Several existing works study either adversarial or natural distributional robustness of deep neural networks separately. In practice, however, models need to enjoy both types of robustness to ensure reliability. In this work, we bridge this gap and show that in fact, *explicit tradeoffs* exist between adversarial and natural distributional robustness. We first consider a simple linear regression setting on Gaussian data with disjoint sets of *core* and *spurious* features. In this setting, through theoretical and empirical analysis, we show that (i) adversarial training with ℓ_1 and ℓ_2 norms increases the model reliance on spurious features; (ii) For ℓ_∞ adversarial training, spurious reliance only occurs when the scale of the spurious features is larger than that of the core features; (iii) adversarial training can have *an unintended consequence* in reducing distributional robustness, specifically when spurious correlations are changed in the new test domain. Next, we present extensive empirical evidence, using a test suite of twenty adversarially trained models evaluated on five benchmark datasets (ObjectNet, RIVAL10, Salient ImageNet-1M, ImageNet-9, Waterbirds), that adversarially trained classifiers rely on backgrounds more than their standardly

trained counterparts, validating our theoretical results. We also show that spurious correlations in training data (when preserved in the test domain) can *improve* adversarial robustness, suggeting that prior claims that adversarial vulnerability is rooted in spurious correlations are incomplete.

4.1 Introduction

Despite continuously improving upon state of the art accuracy on various benchmarks, deep image classifiers remain brittle to distribution shifts, suffering massive performance drops when evaluated on non-i.i.d. data. For example, the accuracy of object detectors trained on ImageNet [14] reduces by 40-45% on ObjectNet [37], where images are taken within households at various viewpoints and rotations. The reliance of deep models on *spurious features*, which correlate with class labels but are irrelevant to the true labeling function [76], is one cause of poor model robustness, as performance degrades when spurious correlations are broken. Indeed, model reliance on spurious features like texture [85] and background [11, 39] is well documented. Of greater concern, deep models in safety critical applications such as detection of pneumonia [92] and COVID-19 [69] have been observed to rely on hospital specific spurious markers, causing poor generalization to new hospitals.

Adversarial examples [4, 93] pose another troubling distribution shift, where imperceptible input perturbations can cause model accuracy to drop to zero. Many works have been proposed to improve the adversarial robustness of deep models [94, 95, 96, 97, 98, 99, 100, 101], including the widely popular *adversarial training*, where inputs are augmented via adversarial attack during training [60]. While spurious correlation robustness has also attracted lots of attention [71, 72, 76, 79], the two problems are most often considered independently, despite both being essential

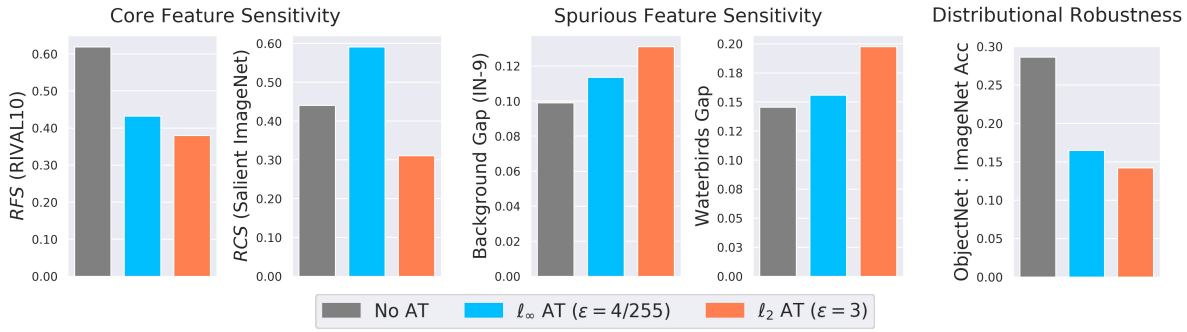


Figure 4.1: Snapshot of empirical evidence using *RIVAL10, Salient ImageNet-1M, ImageNet-9, Waterbirds,* and *ObjectNet* benchmarks. Results averaged over ResNet18 and ResNet50. **Adversarial training, especially under ℓ_2 norm, reduces (increases) sensitivity to core (spurious) features. The increased reliance on spurious features leads to worse distributional robustness**.

to the safe and reliable deployment of deep models in the wild.

In the few works that do consider adversarial and spurious correlation robustness in tandem, the prevailing argument is that the origin of adversarial vulnerability is that the model focuses on spurious correlations that can be manipulated [102, 103, 104]. However, recently, noise-based analyses on RIVAL10 [15] and Salient ImageNet-1M [74] datasets suggest adversarial training may actually increase model sensitivity to spurious features; a result that is both counter-intuitive and in direct contrast to existing ideology.

To better understand this observation, we first appeal to a simple linear regression problem on Gaussian data with disentangled *core* and *spurious* features. In this setting, we theoretically show

- Adversarial training under ℓ_2 and ℓ_1 norms increase model reliance on spurious features, as using spurious features forces an attacker to spread its budget over additional features.

- For ℓ_∞ adversarial training, increased spurious feature reliance only occurs when *the scale* of the spurious feature is larger than that of the core features. That is, spurious features

are used when perturbations that corrupt core features are too small to disrupt spurious correlations.

- Due to increased spurious feature reliance, there is an *explicit tradeoff* between adversarial and distributional robustness. Specifically, we show that adversarial training decreases model robustness to distribution shifts in the test domain where spurious correlations are broken.

To validate our theory, we evaluate twenty models adversarially trained using ℓ_2 and ℓ_∞ projected gradient descent [105]. Specifically, we inspect performance on multiple spurious robustness benchmarks over synthetic (ImageNet-9 [39], Waterbirds [72]) and real (RIVAL10 [15], Salient ImageNet-1M [74], ObjectNet [37]) datasets. Figure 4.1 summarizes our experiments, where we find that adversarially trained models consistently show greater sensitivity to spurious features compared to standardly trained baselines, with the effect more dramatic for ℓ_2 adversarial training than ℓ_∞[1]. Finally, we show that the presence of spurious correlations in training data (when preserved in test domain) can improve adversarial robustness, with stronger spurious correlations leading to greater accuracy under attack.

Our work combines two prevalent but often separately considered notions of robustness, yielding surprising theoretically-derived and empirically-supported results. We hope our contributions grant insight to both adversarial and distributional robustness communities, and emphasize the need for holistic evaluations of model robustness.

[1]High RCS for ℓ_∞ AT models is due to reduced *scale* of contextual bias in Salient ImageNet since the data diversity weakens background correlations. In ℓ_1 and ℓ_2 adversarial training, models rely on spurious features regardless of their scale. See details in Section 4.3.

4.2 Review of Literature

Adversarial Robustness. Since adversarial examples were first observed in deep models [4, 93], the phenomenon has been extensively studied. New attacks [106, 107, 108, 109] and defenses [94, 110, 111, 112] are introduced frequently, in a game of cat and mouse where the attacker generally has the upper hand [113]. Certified defenses seek to break this cycle by offering provable robustness guarantees [96, 97, 98, 99]. Arguably the most popular defense is *adversarial training* [60] where images are augmented with adversarial perturbations during training, amounting to a min max optimization.

Natural Distributional Robustness. In contrast to synthetic adversarial perturbations, many works seek to characterize the robustness of deep models to naturally occurring distribution shifts, for instance due to common corruptions (noise, blur, etc) [35] or changes in rendition [84, 114]. [73] compiles ten benchmarks of realistic distribution shifts over diverse applications (medical, economic, etc). Many algorithms have been proposed to improve out-of-distribution robustness [71, 72, 79, 115], though in comprehensive evaluations, their gains over empirical risk minimization are marginal, as they often only hold for *certain* distribution shifts [83, 84]. [83] identifies *diversity* and *correlation* shifts as two key dimensions to OOD robustness; our work focuses on the latter.

Spurious Correlations. Solely optimizing for accuracy leads deep models to rely on *any* patterns predictive of class in the training domain. This includes *spurious* features, which are irrelevant to the true labeling function. Natural image datasets are riddled with spurious features [1, 116]. Spurious feature reliance becomes problematic under distribution shifts that break their correlation to class labels: sidewalk segmentation struggles in the absence of cars [6], familiar objects cannot

be recognized in unfamiliar poses [117] or uncommon settings [37, 87], etc. A natural and ubiquitous spurious correlation in vision is image backgrounds, observed in numerous prior works to be leveraged by models for classification [15, 39] and object detection [7]. Spurious correlations also relate to algorithmic biases [85, 118], with implications for fairness [9, 53, 119, 120], reflecting the importance of this issue.

Accordingly, many works seek to improve spurious correlation robustness. Families of approaches include optimizing for worst group accuracy [40, 72, 78, 79, 121], learning invariant latent spaces [71, 80], appealing to meta-learning [81] or causality [82, 122]. Our work does not focus on mitigation methods, but instead sheds insight on how optimizing for a different notion of robustness (i.e. adversarial) affects spurious feature reliance, and consequently, natural distributional robustness. Generally, models trained under ERM are believed to have a propensity to use spurious features, especially when they are easy to learn, due to bias of learners (algorithmic and human) to absorb simple features first [77] and take shortcuts [5]. However, recent work suggests that core features are still learned under ERM even when spurious features are favored, and simple finetuning on data without the spurious feature can efficiently reduce spurious feature reliance without full model retraining [76].

Unintended Outcomes of Adversarial Training. Adversarial training achieves improved accuracy under attack, but comes at the cost of standard accuracy, with multiple works provably demonstrating this tradeoff [95, 123, 124]. Notably, [125] inspires our theoretical analysis, though we focus on the effect of adversarial training on out-of-distribution (OOD) robustness to spurious correlation shifts, rather than its effect on standard accuracy. A more positive outcome is that adversarial training leads to perceptually aligned gradients [126], with applications to model debugging [42, 48], and further, transfer learning on adversarially robust features (despite

being less accurate on the original task) yields better accuracy on downstream tasks compared to features learned from standard training [105].

To our knowledge, robustness to adversarial and natural distribution shifts have not been studied in tandem. However, spurious correlations are at times mentioned with adversarial robustness, usually in claims that the origin of adversarial vulnerability is in model's focus on (imperceptible) spurious features [102, 103, 104]. Our results create tension with the contrapositive of their argument, as we show that mitigating adversarial vulnerability (via adversarial training) results in *increased* spurious feature reliance. We do this analytically in a simple linear regression setting (Section 4.3), and empirically on multiple benchmarks, with an emphasis on natural spurious features (i.e., backgrounds) in our experiments (Section 4.3). Further, we even demonstrate a case where the presence of a spurious feature leads to *improved* adversarial robustness (Section 4.4.3)). We note that the spurious features we observe to be positively associated with adversarial robustness may be distinct from those that prior works claim contribute to adversarial vulnerability. However, our result of adversarial training leading to increased spurious feature reliance (of any kind) is novel and contrary to common understanding. Given the critical nature of adversarial and spurious correlation robustness for model security, reliability, and fairness, the significance of our result in revealing potential misconceptions on the interplay of these two crucial modes of robustness should not be understated.

[15] and [74] recently observed decreased core sensitivity on a handful of ℓ_2 adversarially trained models, in spirit with our findings, but with no explanation. We offer the first rigorous analysis of this counterintuitive phenomenon, evaluating 5 to 10 times as many models in 5 times as many settings. More importantly, we theoretically prove that adversarial training increases spurious feature reliance, contributing novel fundamental insight as to how optimizing for adver-

sarial robustness can lead to reduced robustness to natural distribution shits, uncovering important effects like the norm of adversarial training and the scale of spurious features at play.

4.3 Theoretical Analysis on Linear models

We begin by analysing the effects of adversarial training on a simple linear regression model. Consider the model $Y = \langle X, \theta^{\text{opt}} \rangle + W$ where $X \in \mathbb{R}^m$ is the input variable, $\theta^{\text{opt}} \in \mathbb{R}^m$ is the optimal parameter, $\langle ., . \rangle$ represents the inner product and $W \in \mathbb{R}$ is a noise variable. We assume that the input variables follow a multivariate Gaussian distribution $N(0, \Sigma)$ where $\Sigma \in \mathbb{R}^{m \times m}$ is the covariance matrix and further assume that W is sampled from the Gaussian distribution $N(0, \sigma_w^2)$.

We assume that the set of features $[m]$ consists of two groups, the *core features* C and the *spurious features* S. Without loss of generality, we assume that $C = \{1, \ldots, p\}$ and $S = \{p+1, \ldots, m\}$. We assume that the optimal parameter θ^{opt} has non-zero entries on the set C only. This implies that the output depends on the input only through the core features and conditioned on the core features, it is independent of the spurious ones. More formally, we assume that $Y \perp X_S | X_C$ where X_S and X_C represent the core and spurious subsets of the input, respectively.

For loss functions, we define the *standard loss* function as $L(\theta) = \mathbb{E}\left[(y - \langle X, \theta \rangle)^2\right]$,, and the *adversarial loss* function as

$$L_{p,\epsilon}(\theta) = \mathbb{E}\left[\max_{\|\delta\|_p \leq \epsilon} (Y - \langle X + \delta, \theta \rangle)^2\right], \tag{4.1}$$

where p is the attack norm and ϵ represents the norm budget.

We first show an equivalent form of (4.1) that is more amenable to analysis.

Theorem 1. *Assume that* $Y = \langle X, \theta^{opt} \rangle + W$ *where* $W \sim N(0, \sigma_w^2)$ *is independent of* X *and* $\theta^{opt} \in \mathbb{R}^m$ *is a fixed parameter. Assume further that* X *follows the distribution* $N(0, \Sigma)$ *and define* σ_θ^2 *as* $(\theta - \theta^{opt})^T \Sigma (\theta - \theta^{opt}) + \sigma_w^2$. *The loss function* (4.1) *is equivalent to*

$$L_{p,\epsilon}(\theta) = c_2 \cdot \sigma_\theta^2 + (c_1 \sigma_\theta + \epsilon \cdot \|\theta\|_q)^2 \tag{4.2}$$

where $c_1 = \sqrt{\frac{2}{\pi}} < 1$, $c_2 = 1 - c_1^2$ *and* $\|.\|_q$ *is the dual norm of* $\|.\|_p$, *i.e,* $\frac{1}{p} + \frac{1}{q} = 1$. *Furthermore, the above formulation is convex in* θ.

The above result is similar to Proposition 3.2 in [125] which provides characterization results for the ℓ_2 norm. The key distinction of our results is providing a simple convex formulation of the robust minimization problem, allowing the results to be easily generalized for an arbitrary ℓ_p norm. The theorem shows that the optimal value $\widehat{\theta}$ minimizing the adversarial loss $L_{p,\epsilon}(\theta)$ *is not* θ^{opt} and in general, may be non-zero on the set of spurious features S. This means that adversarial training directs the model towards using the spurious correlations in order to increase robustness.

The proof of the theorem is provided in the Appendix. The main structure of the proof is similar to that of [125]. We first show that the inner maximization problem of (4.1) can be rewritten as

$$\max_{\|\delta\| \leq \epsilon} (Y - \langle X + \delta, \theta \rangle)^2 = \Big(|Y - \langle X, \theta \rangle| + \epsilon \cdot \|\theta\|_q \Big)^2 . \tag{4.3}$$

This allows us to rewrite (4.1) as

$$
\begin{aligned}
L_{p,\epsilon} &= \mathbb{E}\left[\left(|Y - \langle X, \theta\rangle| + \epsilon \cdot \|\theta\|_q\right)^2\right] \\
&= \mathbb{E}\left[(Y - \langle X, \theta\rangle)^2\right] + \epsilon^2 \cdot \|\theta\|_q^2 + 2 \cdot \epsilon \cdot \|\theta\|_q \cdot \mathbb{E}\left[|Y - \langle X, \theta\rangle|\right] \\
&\overset{(a)}{=} \mathbb{E}\left[(\langle X, \theta - \theta^{\text{opt}}\rangle + W)^2\right] + \epsilon^2 \cdot \|\theta\|_q^2 + 2 \cdot \epsilon \cdot \|\theta\|_q \cdot \mathbb{E}\left[|\langle X, \theta - \theta^{\text{opt}}\rangle + W|\right],
\end{aligned}
$$

where for (a) we have used the fact that $Y = \langle X, \theta\rangle + W$. Using the fact that X and W are Gaussian, we show that this is equal to $\sigma_\theta^2 + \epsilon^2 \cdot \|\theta\|_q^2 + 2 \cdot c_1 \cdot \epsilon \cdot \|\theta\|_q \cdot \sigma_\theta$ which we can further simplify to (4.2) with some algebraic manipulation. Next, using standard techniques for analysing composition of convex functions, we show that σ_θ^2 and $(c_1\sigma_\theta + \epsilon \cdot \|\theta\|_q)^2$ are both convex in θ, implying that $L_{p,\epsilon}(\theta)$ is convex in θ as well.

Using the convex formulation in Theorem 1, we evaluate the linear model for different values of parameters to better understand the reliance of the model on spurious features.

In our experiments, we consider a simple model with 5 features, the first two of which are core features.

We let η be a parameter controlling the correlation degree between the core and spurious features, with larger values corresponding to higher correlation. For the distribution of the data, define the

matrix $\widetilde{Q}$ and Q matrices as

$$\widetilde{Q} = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 & 0 \\ \eta & \eta & 1 & 0 & 0 \\ \eta & \eta & 0 & 1 & 0 \\ \eta & \eta & 0 & 0 & 1 \end{bmatrix}, \quad Q_{i,j} = \frac{\widetilde{Q}_{i,j}}{\sqrt{\sum_{i,j'} \widetilde{Q}^2_{i,j'}}} \tag{4.4}$$

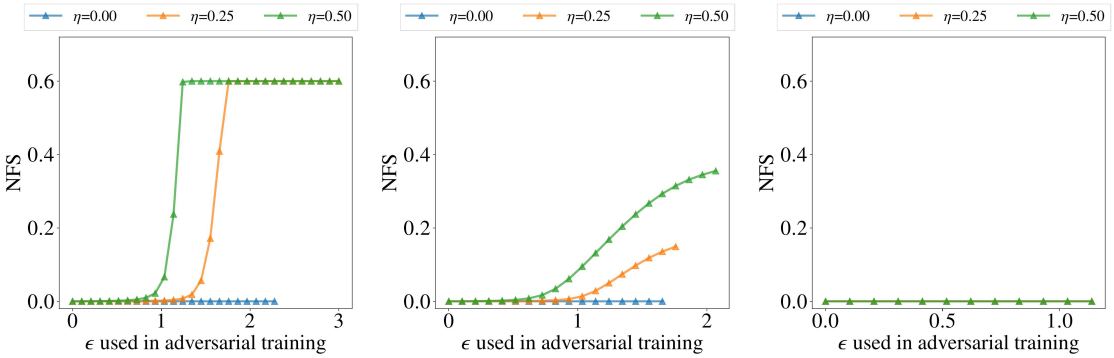
Note that Q is obtained by normalizing the rows of $\widetilde{Q}$. Each row of the Q matrix corresponds to an input feature. We let Σ take the value QQ^T. This is equivalent to sampling X from the distribution $Q\mathcal{N}(0, I)$. Throughout our experiments, we set $\sigma_w = 0.1$.

For an arbitrary vector θ, we define its Norm Fraction over Spurious features NFS(θ) as

$$\text{NFS}(\theta) = \frac{\sum_{i \in S} \theta_i^2}{\sum_j \theta_j^2} \tag{4.5}$$

Intuitively, NFS measures the degree to which a model relies on spurious features. For our first experiment, we perform adversarial training with varying parameter ϵ to obtain a predictor $\widehat{\theta}$ and evaluate its NFS. We consider different choices of the ℓ_p norm as well as the correlation parameter η The results can be seen in Figure 4.2.

As seen in Figure 4.2, for the ℓ_1 and ℓ_2 norms, increasing the adversary's budget causes the model to rely more on the spurious features. To understand why this happens, it is helpful to consider a game theoretic perspective: If the model only looks at the core features, then the adversary will only need to perturb these features. Thus, even though the spurious features are normally less suitable for prediction, they now have the advantage of being less perturbed. Assuming ϵ is large



(a) ℓ_1 norm (b) ℓ_2 norm (c) ℓ_∞ norm

Figure 4.2: Reliance of the adversarially trained model on spurious features as measured by NFS (see Equation (4.5)) for different choices of ℓ_p norm and different values of adversarial budget ϵ and spurious correlation parameter η.

enough, the model will be better off looking at these features in forming its prediction. Of course, if the model only looks at the spurious features, core features will become even more informative as they would be unperturbed as well. In the game's equilibrium, the model would use both the core and spurious features, relying more on the spurious features with increased values of ϵ.

Interestingly, this reasoning does not always apply for the ℓ_∞ norm. Indeed, if the model were to only look at the core features, the adversary may still perturb the spurious features with no extra cost. This is because the ℓ_∞ norm only measures the *maximum* perturbation in each feature and as long as the perturbations on the spurious features are not larger than the perturbations on the core features, the norm would not change. The results of Figure 4.2c support this as the adversary does not rely on the spurious features for its predictions.

Importantly, however, in some cases the adversary may require a larger budget to perturb the spurious features compared to core features. Indeed, if we were to scale a feature by multiplication with a large number, then the adversary would require a larger budget to perturb that feature as the budget for that feature has effectively decreased. We can thus expect that for the ℓ_∞ norm, scaling up the spurious features would cause the model to rely on them for prediction.

Figure 4.3 shows that this is indeed the case. The figure shows the norm fraction over spurious

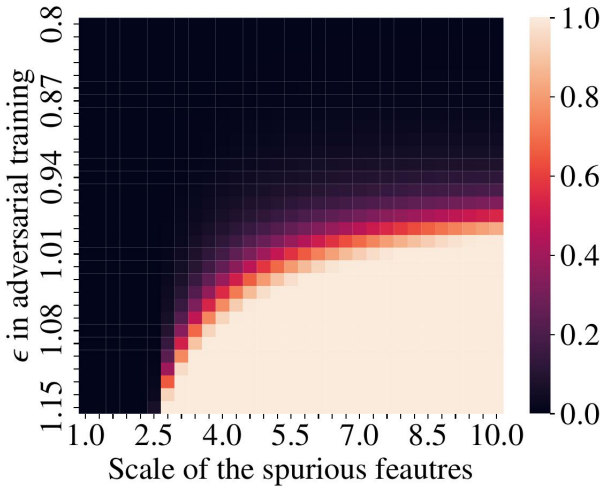
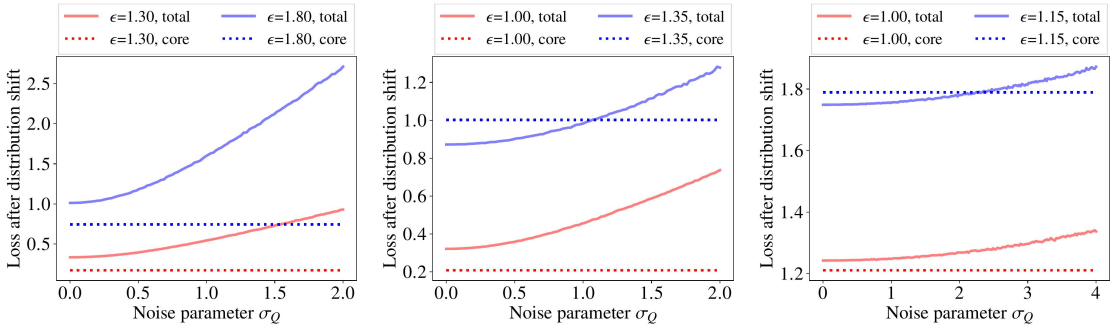


Figure 4.3: Norm fraction over spurious features (NFS) with an ℓ_∞ constrained adversary as a function of the scale of the spurious features and perturbation budget ϵ.

(NFS) after scaling the spurious features for different values of ϵ. The scaling is done by multiplying the rows of Q corresponding to the spurious features by a scaling parameter. As seen in Figure 4.3, larger values of the scaling parameter, as well as larger values of ϵ, cause the model to become more reliant on the spurious features.

Next, we measure the effect of using spurious features on the model's distributional robustness. To do this, we train two adversarial models. The first model, which we denote by "core", uses only the core features while the second model, denoted by "total", uses all of the features. We then simulate a distribution shift that breaks the spurious correlations by adding random Gaussian noise to the spurious rows of matrix Q defined in (4.4). Specifically, for each entry of Q in a spurious row, we add a noise sampled from $N(0, \sigma_Q^2)$ where σ_Q is a noise parameter. We use a fixed value of $\eta = 0.25$ and vary the parameter ϵ, the norm p used in adversarial training as well as the variance of the Gaussian noise. The results are shown in Figure 4.4.



(a) ℓ_1 norm (b) ℓ_2 norm (c) ℓ_∞ norm

Figure 4.4: Effect of reliance on spurious features on distributional robustness. Each figures compares two models, one using only the core features, and another using all of the features (denoted by "core" and "total" respectively). For the ℓ_1 and ℓ_2 norms, the scale of the spurious features is 1 while for the ℓ_∞ norm, the scale is set to 3.

We observe that the clean (i.e. in distribution; $\sigma_Q = 0$) loss of the "core" model may be higher or lower than that of the "total" model, but in both cases, the total model is consistently more vulnerable to distributional shifts resulted from breaking spurious correlations.

4.4 Empirical Evidence

We now demonstrate increased spurious feature reliance in adversarially trained models over multiple benchmarks. We evaluate models on two backbones (ResNet18, ResNet50) adversarially trained on ImageNet [14] using two norms (ℓ_2, ℓ_∞) under five attack budgets (denoted ϵ) per norm, resulting in a $2 \times 2 \times 5 = 20$ model test suite, as well as standardly trained baselines. See appendix for details.

4.4.1 AT hurts Natural Distributional Robustness *only* when Spurious Correlations are broken

We first show reduced distributional robustness of adversarial models occurs specifically in cases where natural spurious correlations are broken. We appeal to the **ImageNet-C** [35] and

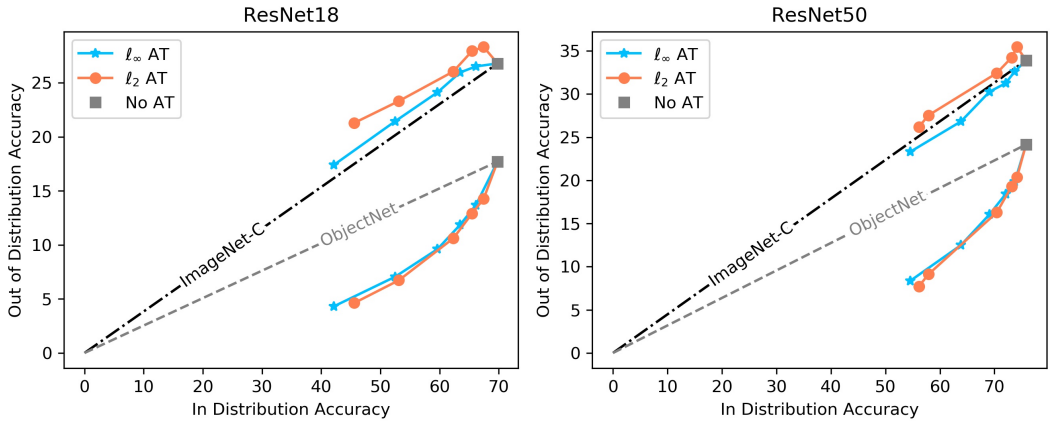


Figure 4.5: OOD accuracy vs standard ImageNet accuracy for adversarially trained ResNets. ImageNet-C accuracy is closely tied to standard accuracy, but for ObjectNet, where spurious correlations are broken, performance drop is more severe than a linear relation with standard accuracy would entail.

ObjectNet [37] OOD benchmarks. ImageNet-C augments ImageNet samples with common corruptions like noise or blurring, distorting both core and spurious features equally. Crucially, these corruptions do not break spurious correlations. On the other hand, ObjectNet is formed by having workers capture images of common household objects (including samples from 113 classes of ImageNet) *in their homes*. Thus, only spurious features are affected. Namely, ObjectNet introduces distribution shifts in background, rotation, and viewpoint. We plot accuracies on these benchmarks in figure 4.5.

Recall that adversarially trained models have lower standard accuracy, which can confound our analysis, so we compare the drop in OOD accuracy to the drop in ImageNet accuracy across our model suite. Observe that the ratio of ImageNet-C accuracy to ImageNet accuracy is roughly constant across models. However, the ratio of ObjectNet accuracy to ImageNet accuracy is *lower* for adversarially trained models. Therefore, even after controlling for reduced standard accuracy, the distributional robustness of adversarially trained models is worse than that of standard models. Importantly, this effect does not hold for distribution shifts that maintain spurious correlations,

indicating that the reduced distributional robustness is due to increased spurious feature reliance.

4.4.2 Reduced Core Sensitivity, and Difference in the Effect of ℓ_2 & ℓ_∞ Adversarial Training

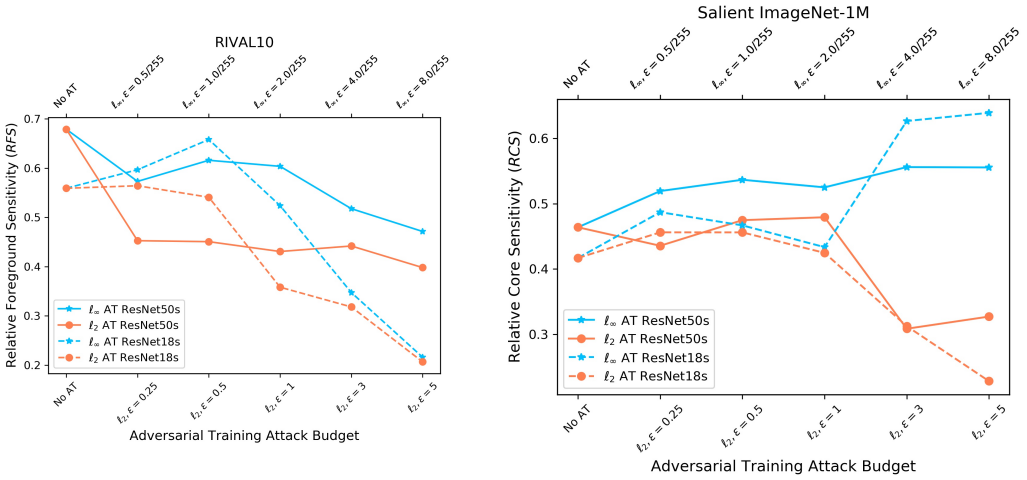


Figure 4.6: Noise-based evaluation of model sensitivity to foreground (RFS on RIVAL10) or core (RCS on Salient ImageNet-1M) regions. Lower values entail greater sensitivity to spurious regions.

We now directly quantify sensitivity to core features via **RIVAL10** and **Salient ImageNet-1M** datasets [15, 74]. The premise of this analysis is that model sensitivity to an input region can be quantified by the drop in accuracy due to corrupting that region [1]. [15] introduced the noise-based metric *relative foreground sensitivity* (RFS), which is the gap between accuracy drops due to background and foreground noise, normalized so to allow for comparisons across models with varying general noise robustness. RIVAL10 object segmentations allow for RFS computation. Analogously, *relative core sensitivity* (RCS) is computed using Salient ImageNet-1M's soft segmentations of core input regions. A key distinction between the two metrics is that RCS is computed directly on pretrained models performing the original 1000-way ImageNet

classification task, while RFS first requires models to be finetuned on the **much coarser** 10-way classification task of RIVAL10. Also, Salient ImageNet-1M includes *all ImageNet images*, while RIVAL10 only consists of 20 ImageNet classes.

Figure 4.6 shows a decrease in RFS and RCS as the attack budget ϵ seen during ℓ_2 adversarial training rises. Thus, **adversarial training reduces core feature sensitivity relative to spurious feature sensitivity**. Notably, this effect does *not* hold for models adversarially trained with attacks under the ℓ_∞ norm for RCS, though it does for RFS. Alluding to our theoretical result, we conjecture that in Salient ImageNet-1M, the scales of the spurious features are much smaller than in RIVAL10, due to the diversity of images and finer grain of classes. That is, a smaller perturbation is needed to alter a spurious feature so that it correlates with an incorrect class when there are 1000 classes than when there are only 10 classes with generally disparate backgrounds.

4.4.3 Adversarial Training Increases Background Reliance in Synthetic Datasets

Now, we take a closer look at the reliance of adversarially trained models on the contextual spurious feature of *backgrounds* via the synthetic datasets **ImageNet-9** [39] and **Waterbirds** [72]. Both datasets use segmentations to superimpose objects over varying backgrounds, detailed below.

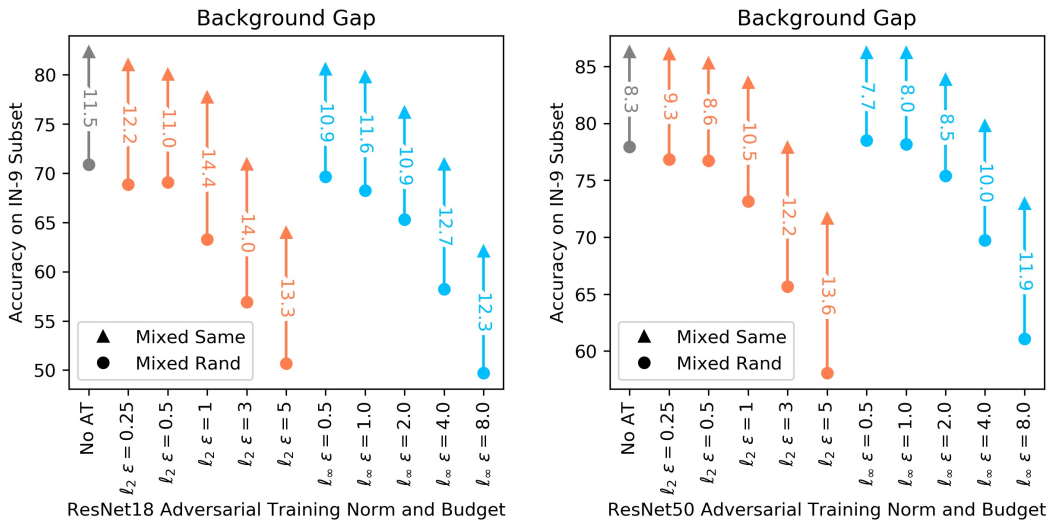


Figure 4.7: Background Gap (difference in accuracies on ImageNet-9 subsets MIXED-SAME and MIXED-RAND). The drop in accuracy due to background cross-class swapping (MIXED-RAND) causes larger drops in accuracy for robust models, especially ℓ_2 adversarially trained models.

ImageNet-9 (IN-9) organizes a subset of ImageNet into nine superclasses. Multiple validation sets exist for IN-9, where backgrounds or foregrounds are altered; we use MIXED-SAME and MIXED-RAND. In both sets, original backgrounds are swapped out for new ones. Crucially, in MIXED-SAME, the new backgrounds are taken from other instances *within the same class*, while MIXED-RAND uses *random backgrounds*. The metric, **Background Gap**, is the difference in model accuracy on MIXED-SAME and MIXED-RAND (i.e. drop due to breaking spurious background correlation).

Figure 4.7 shows that the background gap for our test suite of twenty robust ResNets and two standardly trained baselines. Nine out of the ten ℓ_2 adversarially trained models have larger background gaps than the standard baselines, while the same is true for five out of the ten ℓ_∞ adversarially trained models. When considering relative gaps (i.e. as a percent of the accuracy on MIXED-SAME), the increase in gap becomes even more dramatic, with the ℓ_2 adversarially trained ResNet50 for $\epsilon = 5$ having a 18.9% relative drop, compared to the 9.7% relative drop for

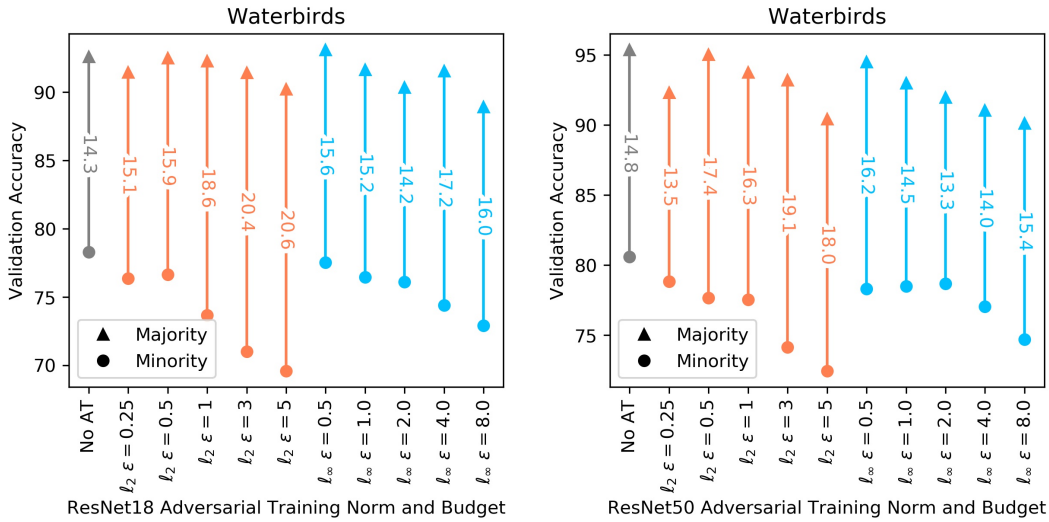


Figure 4.8: Accuracies on Waterbirds subsets where spurious correlation is intact (majority group; e.g. land birds on land backgrounds) and where it is broken (minority group; e.g. land birds on water backgrounds). The drop in performance due to breaking the background correlation grows larger for robust models, especially ℓ_2 adversarially trained models.

the standardly trained ResNet50.

Waterbirds combines foregrounds from Caltech Birds [30] and backgrounds from SUN Places [127]. The task is binary classification of land birds and water birds. The majority group (95% of training samples) consists of land birds over land backgrounds and water birds over water backgrounds. The minority group breaks this spurious correlation, placing land birds over water backgrounds, and vice versa. The test set is evenly split between these groups. We train only a final linear layer atop the frozen feature extractors (so that models remain adversarially robust) for each of our models on the Waterbirds training set for ten epochs, saving the model with highest validation accuracy.

Figure 4.8 shows majority and minority group accuracies, and the gap between them, for our test suite of models. Again, we see increased gaps for robust models, with 100% of ℓ_2 and 60% of ℓ_∞ models respectively having larger gaps than the standardly trained baseline on the corresponding backbone.

In both benchmarks, breaking the background spurious correlation causes a more significant drop in performance for adversarially trained models than standardly trained models, indicating that adversarial training led to increased reliance on backgrounds. The observed affect is stronger for ℓ_2 adversarially trained models than ℓ_∞ ones. Further, the gaps grow near monotonically with ϵ.

4.4.4 Reverse Effect: Presence of Spurious Correlations Can *Improve* Adversarial Robustness

Finally, we show evidence that is directly at odds with the claim that spurious features lead to adversarial vulnerability. We train ResNet18s on CIFAR10 [26] with a spurious feature injected. Namely, images have all values in one color channel slightly increased. The majority group consists of red-shifted images from classes $0-4$ and green shifted images from classes $5-9$, while the minority group has reverse color-shifts. The parameter ρ is the ratio between majority and minority group size, controlling the strength of the spurious correlation (higher ρ means stronger spurious correlation; $\rho = 1:1$ means the spurious feature has no predictive power). We then evaluate the accuracy of the trained model under adversarial attack on an i.i.d. test set (**spurious feature retained**, no distribution shift).

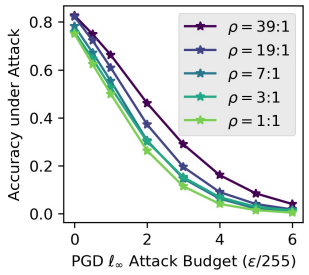


Figure 4.9: Injecting spurious feature improves adversarial robustness.

Figure 4.9 visualizes the results. Not surprisingly, the clean accuracy is higher for models trained on data with higher ρ, as the spurious feature is more predictive for higher ρ. In fact, the added predictive influence the spurious feature leads to better accuracy *under attack*, with the

gap between highest and lowest ρ values growing up to four fold compared to the baseline gap in clean accuracy. Thus, using a spurious feature can improve adversarial robustness. Despite being on a contrived example, this experiment shows that, while some spurious correlations may cause adversarial vulnerability, *others do the opposite*: the picture is more nuanced than previously assumed.

Part II: Scalable Discovery and Mitigation of Deficiencies

Having observed unexpected model limitations via manually crafted datasets, we now explore the possibilities of more scalable analyses. Across this chapter, I will present multiple cases where pretrained models can be leveraged in auxiliary fashion to enhance transparency around the inner workings of models and their failure modes. The first two chapters in this part utilize the embedding spaces of contrastively-trained models, first to efficiently catch and categorize adversarial attacks, and then to allow for open-vocabulary access to activation spaces of arbitrary vision encoders by extending CLIP text encoders to much smaller models trained on far less data. Both of these results only require training linear layers, indicating that model collaboration can be achieved with minimal additional compute. Building on these successes, the later three chapters utilize rationales generated from frontier models, attributes from image taggers, and neural activations from adversarially trained models respectively, with auxiliary models used directly off-the-shelf, i.e. with no additional training required. In all three cases, the auxiliary model is primarily used to label or organize subsets of data. Then, by inspecting the difference in model outputs for these curated subsets, we can identify performance disparities indicative of a bias or vulnerability. Essentially, data serves as a mutual language between models and the humans who wish to understand them. Moreover, these same subsets of data can often be instrumental in fixing the very limitations they uncover. In summary, collaborating with models to better understand the data we already have is a promising approach to shed light on the use-cases a model may encounter and the specific cases where it may struggle the most. This finding offers a nuanced perspective on the notion of transparency this dissertation advocates for: we need not understand every detail of how a model operates; rather, simply knowing when a model may struggle can improve reliability, as practitioners can refrain from using the model in those instances, and developers can focus their efforts on mitigating the surfaced failures.

Chapter 5: Sample Efficient Detection and Classification of Adversarial Attacks via Self-Supervised Embeddings

This chapter details an ICCV '21 paper co-authored with Soheil Feizi [111]. In it, we leverage a surprising property of self-supervised embeddings to tackle the issue of adversarial robustness of image classifiers.

Adversarial robustness of deep models is pivotal in ensuring safe deployment in real world settings, but most modern defenses have narrow scope and expensive costs. In this paper, we propose a self-supervised method to detect adversarial attacks and classify them to their respective threat models, based on a *linear* model operating on the embeddings from a pre-trained self-supervised encoder. We use a SimCLR encoder in our experiments, since we show the SimCLR embedding distance is a good proxy for human perceptibility, enabling it to encapsulate many threat models at once. We call our method **SimCat** since it uses SimCLR encoder to catch and categorize various types of adversarial attacks, including ℓ_p and non-ℓ_p evasion attacks, as well as data poisonings. The simple nature of a linear classifier makes our method efficient in both time and sample complexity. For example, on SVHN, using only five pairs of clean and adversarial examples computed with a PGD-ℓ_∞ attack, SimCat's detection accuracy is over 85%. Moreover, on ImageNet, using only 25 examples from each threat model, SimCat can classify eight different attack types such as PGD-ℓ_2, PGD-ℓ_∞, CW-ℓ_2, PPGD, LPA, StAdv, ReColor, and JPEG-ℓ_∞,

with over 40% accuracy. On STL10 data, we apply SimCat as a defense against poisoning attacks, such as BP, CP, FC, CLBD, HTBD, halving the success rate while using only twenty total poisons for training. We find that the detectors generalize well to unseen threat models. Lastly, we investigate the performance of our detection method under adaptive attacks and further boost its robustness against such attacks via adversarial training.

5.1 Introduction

Deep learning has been applied to many applications with great success, though a major roadblock to safe deployment of deep models in real world settings is their susceptibility to adversarial attacks. Targeting a model at the time of inference is known as an evasion attack [4], where imperceptible perturbations are made to an input to craft an *adversarial example* that is misclassified by the model. Models can also be attacked during training via poisoning [128], where a small number of adversarial examples are inserted to a training set, so that after training, targeted test samples are misclassified.

To mitigate these vulnerabilities, many defenses have been introduced [60, 94, 99, 129]. However, a number of practical challenges remain. First, most defenses only harden a model against a specific narrow threat model [130]. Thus, an attacker can easily evade the defense by using a different attack. Second, novel attacks are introduced frequently, in a brisk game of cat and mouse, in which the attacker generally has the upper hand [113]. When we combine the rapid development of attacks with the high computational cost of retraining a model to be robust against even a single threat, maintaining the robustness of a model to all threats using typical defenses becomes intractable.

Certain detection based defenses [131] do not require fundamentally changing how the model is trained, which allows for easy application of the detector without disrupting the existing machine learning pipeline. However, many detection based systems require large amounts of data. This is problematic in practice, as an adversary may employ an attack that the defender has never seen before, making it difficult for the defender to have trained the detector to work against it. Even if the defender obtains some examples of the novel attack, the training size may not be sufficient to expand the detector's ability to include the novel attack.

The advantage of a detector that is both broad in scope and inexpensive in training requirements is clear. In this paper, we propose a highly sample efficient detector that also generalizes well to unforeseen attacks. Further, we extend our model to classify adversarial attacks to their respective threat models, similar to [132]. The classification allows for additional defenses to be employed, specific to the attack encountered. Our model is based on pretrained self-supervised encoders, which are capable of efficiently extracting the semantic content of images, as evident by recent successes in the self-supervised domain that have closed the gap with supervised models.

Specifically, we use a SimCLR encoder pretrained on ImageNet, because we observe that distance in the embedding space of this encoder correlates strongly with human perception (Figure 5.1). Furthermore, the distance between the SimCLR representations of a clean image and its adversarially perturbed counterpart is similar across various threat models, making the SimCLR distance a strong candidate to proxy the true perceptual threat model, which encapsulates all imperceptible adversarial attacks [94]. While LPIPS [133] has also been shown to be a strong proxy for true perceptual distance, the dimensionality of its feature vectors is massive in comparison to that of SimCLR representations. Our method is called **SimCat**, as it uses a SimCLR encoder to linearly catch adversarial attacks and categorize them to their respective threat models.

By utilizing the highly informative low dimensional embedding space of self-supervised encoders, we find that even a linear model trained on these representations can effectively detect and classify adversarial attacks of a wide variety of types. Our experiments over many threat models and multiple datasets show that, for both evasion and poisoning attacks, SimCat greatly outperforms baseline models. While the method is extremely simple, we argue that its simplicity is essential for the efficiency of the method both in terms of time and sample complexities. By freezing the encoder, SimCat's optimization is *convex* and thus its global optimum can be found effectively. Moreover, SimCat has low model complexity due to the small dimensionality of SimCLR representations, allowing for highly efficient training that also generalizes well. These properties lead to an impressive empirical performance of SimCat in detection and classification of various types of adversarial examples using as few as 5 training samples per class.

For example, on ImageNet, using only 5 samples per threat model, SimCat's detection accuracy is 68.5%, improving the baseline method by more than 6.4%. Using the same setup, SimCat's classification accuracy of 8 adversarial attacks including PGD-ℓ_2, PGD-ℓ_∞, CW-ℓ_2 [134], PPGD [94], LPA [94], StAdv [135], ReColor [107], and JPEG-ℓ_∞[130] is 27.1%, improving the baseline performance by 7.7%. Using 25 samples per class, SimCat's gains over baseline method grows to 7.4% and 11.8% for detection and classification problems, respectively.

Interestingly, SimCat can be used to detect and classify various types of poisoning attacks as well, which we then apply as an efficient poison defense. We consider five types of poisonings including bullseye polytope (BP) and convex polytope (CP) [136, 137]. Using two SimCat detectors trained with 10 BP and 10 CP poisons respectively, we construct an ensemble detector to remove any sample that is flagged as poison by either of the individual detectors. The ensemble reduces poison success rate over five types of poisoning attacks from 21.8% to 9.7%. Notably,

the SimCat poisoning defense only reduces clean accuracy by 1%.

Lastly, we develop an adaptive attack that creates adversarial examples to evade SimCat detection. We then design an adversarial training procedure with momentum updates and data augmentation to improve robustness of SimCat against adaptive attacks. On ImageNet, the SimCat detector adversarially trained using 25 samples per threat model achieves 71.7% robustness to a PGD-ℓ_2 ($\epsilon = 2.0$) adaptive attack, a 32% improvement over vanilla SimCat. Moreover, the adversarially trained SimCat improves the clean accuracy as well, from 73.2% to 73.6%.

In summary, we make the following contributions:

- We identify that pre-trained SimCLR embeddings contain valuable information regarding perceptibility of adversarial perturbations. Using this intuition, we develop a sample efficient method for detection and classification of adversarial attacks called SimCat.

- We demonstrate the effectiveness of SimCat in detection and classification of various types of adversarial examples in test time (evasion attacks) and training time (poisoning attacks). SimCat leads to impressive empirical results on the ImageNet scale using as few as five training samples per class.

- We study adaptive attacks against SimCat and develop an adversarial training procedure that dramatically increases its robustness to adaptive attacks while improving its clean accuracy.

5.2 Prior Works

5.2.1 Adversarial Attacks and Defenses

Given an input $x \in X$ with label $y \in Y$ and a classifier $\mathbf{f} : X \to Y$, an adversarial attack $\hat{x}$ satisfies $\mathbf{f}(\hat{x}) \neq y, d(\hat{x} - x) \leq \epsilon$, for some generally small bound ϵ. Here, $d(\cdot, \cdot)$ is a distance metric that defines the threat model (i.e. the space of allowable perturbations to the input to craft the adversarial attack). Threat models using ℓ_2 and ℓ_∞ distance are well studied [60, 106], though attacks that apply spatial transformations, recoloring, frequency domain perturbations [107, 130, 135] are also effective. Ideally, a defense would ensure that any two images that are imperceptible to a human are classified in the same way. This motivates the neural perceptual threat model of [94], which utilizes LPIPS distance as a proxy for the true perceptual distance.

Defenses against adversarial attacks either focus on robust prediction or detection. Adversarial training [60] is the most common method for robust prediction. It operates by crafting and training on adversarial examples, with the ground truth label. While it improves robustness for a specific threat model, the gains do not extend to others. Provable defenses based on smoothing have also been proposed, though they pertain to restricted threat models [99, 138].

Attacks can also be made during training, known as data poisoning [128], where a training set is corrupted so that the trained model misclassifies certain target samples. Clean label poisoning attacks are particularly dangerous and covert, as poisons have the correct label, so the accuracy of the model after training remains high. Two of the strongest clean label poisonings attacks are Convex Polytope (CP) [137] and Bullseye Polytope (BP) [136], which work by making imperceptible changes to a set of baseline images from an intended class so that the features

of the baseline images surround a target image, who at test time is then classified to the class of the baseline images. Naturally, robust prediction based defenses can not be applied to data poisoning.

A number of supervised detection methods have been proposed, based on deep network activations [139, 140], statistical tests [141, 142], local intrinsic dimensionality [143], to name a few. Unsupervised methods based on feature squeezing [144], generative models [145], nearest neighbor search, KL divergence [146], among others, have also been suggested. For a comprehensive review, we refer readers to [131]. Generally, unsupervised methods can be costly to configure and sensitive to noise, while supervised methods require lots of data and often fail to generalize to unseen threats. Many detectors have also been shown to be vulnerable [106]. To the best of our knowledge, there is no detection system that uses as few samples as SimCat.

Classifying adversarial examples to their respective threat models has been explored in [132, 147]. This classification can allow for more specific defenses to be applied off the shelf when appropriate, and also give the defender insight about the attacker.

5.2.2 Self-Supervised Encoders

Recent work has seen self-supervised models advance rapidly and in multiple domains [13, 148]. We focus on SimCLR [12], which is trained using contrastive learning.

Contrastive Learning is a simple yet powerful self-supervised framework for representation learning that has made large strides in closing the gap with supervised learning. The contrastive loss seeks to maximize similarity between representations of two views of an input, and minimize similarity to views of other samples. SimCLR uses this simple framework, along with a

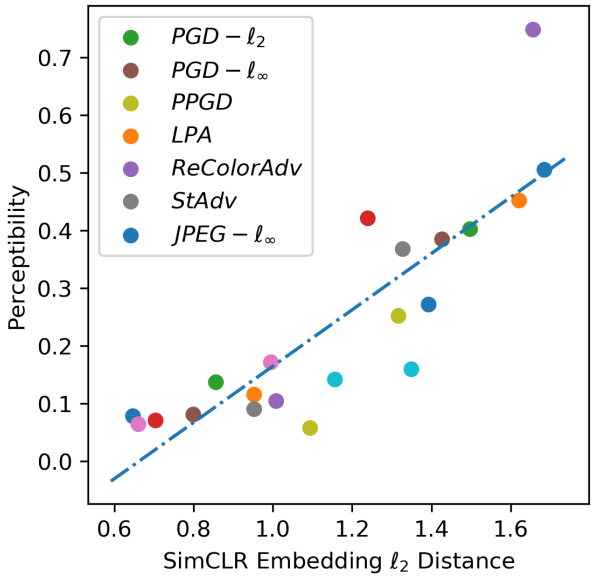


Figure 5.1: Perceptibility of adversarial attacks relative to the ℓ_2 distance between the original and perturbed image in the SimCLR embedding space. Each point in the scatter plot refers to the average distance between adversarial examples within a single threat model under one of three bounds. Correlation is $r = 0.854$.

multi-step data augmentation pipeline for generating different views of the same image, to learn very informative visual representations of images. Specifically, SimCLR indirectly applies the contrastive loss on the representations by way of a shallow MLP projection network appended to the encoder during training, and discarded afterwards. While training self-supervised models can be computationally expensive, in our experiments, we use a *fixed* SimCLR encoder *pre-trained* on ImageNet, available openly from [149].

A few works have looked into adversarially robust contrastive learning [150, 151, 152, 153], though our work differs in that our encoder is fixed and applied to detection and classification of adversarial attacks. Most similar to our work is [154], where SimCLR embeddings are used for anomaly detection. To our knowledge, our work is the first to identify the SimCLR embedding space as one in which adversarial examples and clean images seem to be linearly separable.

5.3 SimCLR distance as proxy for Perceptibility

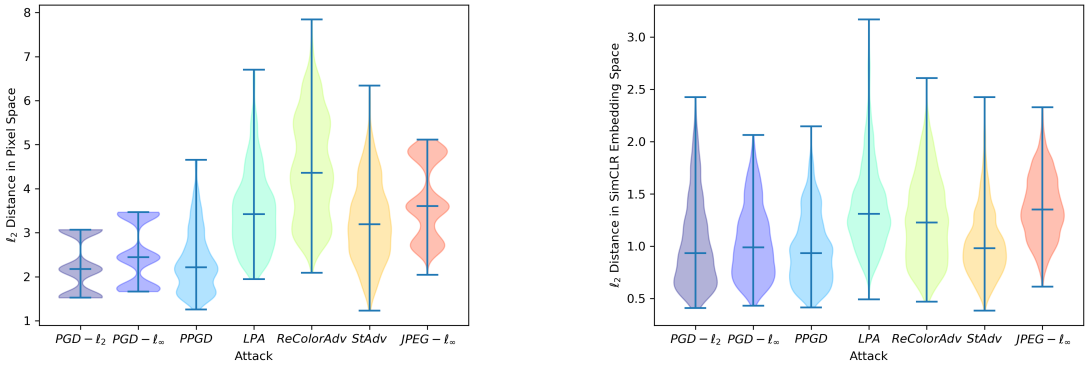


Figure 5.2: The distributions of ℓ_2 distances between adversarial examples and clean images in pixel space (left) and SimCLR embedding space (right). Note the different scales; the distributions across threat models are much more uniform when using SimCLR embeddings.

In this section, we outline the motivation behind using SimCLR as the self-supervised encoder for SimCat. We make use of the data from the human perceptual study in [134]. The data consists of seven threat models, spanning perceptual, ℓ_p, spatial, recoloring, and compression attacks, under three levels of bound on the applied perturbations. Humans were then used to evaluate how perceptible the perturbations were. This was done by presenting a clean and adversarially perturbed sample side by side for two seconds, then having the participant choose whether they thought the images were the same or different. This gives a notion of perceptibly, measured as the ratio of humans who felt the attacked image looked distinct from the original.

In figure 5.1, the mean perceptibility over each threat model and attack bound pair is plotted, against the mean ℓ_2 distance between SimCLR representations of the clean and attacked image. We observe a strong correlation, with Pearson's $r = 0.854$. The correlation is reduced by the high perceptibility of the large coloring attacks. This can be explained by the fact that SimCLR is trained to be less sensitive to color shifts, as color jitter is an important augmentation

employed in the SimCLR pipeline. Removing the coloring attacks, the correlation improves to $r = 0.892$.

Furthermore, SimCLR distance scales similarly for diverse attack types. We observe this in figure 5.2, where some non-ℓ_p attacks require much higher bounds on ℓ_2 distance to be encapsulated. On the other hand, SimCLR distance smoothly distributes the attacks of various types and level. This makes the SimCLR distance a strong proxy for the perceptual threat model, suggesting that it could be useful in adversarial training against unseen threat models, though we leave this to a future work.

We note that the correlation found for ℓ_2 distance in image space and LPIPS are both comparable to the correlation for the SimCLR distance [94]. A key advantage of the SimCLR distance over LPIPS is the low dimensionality of its embeddings. While SimCLR only uses a 2048 dimensional representation vector for each input, LPIPS concatenates flattened feature activations from many layers in a deep network to compute distance, which can lead to a blow up in the size of the representation vector, due to the ever increasing depths and widths of modern deep networks (e.g. for LPIPS evaluated on AlexNet, the representation vector has length upwards of 500,000).

5.4 Methods

5.4.1 General SimCat Framework

In this section, we describe our proposed methods. We use a self-supervised encoder $\phi : \mathbb{R}^D \to \mathbb{R}^d$. Importantly, the self-supervised encoder does not need to be trained or fine-tuned on the data we wish to apply SimCat to. In our experiments, we use a SimCLR encoder with a ResNet50 backbone pretrained on ImageNet to map inputs into a $d = 2048$ dimensional

embedding space. Interestingly, the same encoder can be applied effectively to images of varying size (e.g. SVHN (32), STl10 (96), and ImageNet (224)). We apply a linear transformation on the extracted representations to obtain logits.

For detection, we call our model SimCatch, and denote it by $\mathbf{d}_{\phi,\boldsymbol{\omega}}$, where $\boldsymbol{\omega}$ contains all the $d+1$ trainable parameters, consisting of vector weights $\mathbf{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$. The SimCatch detector maps $\mathbf{d}_{\phi,\boldsymbol{\omega}} : \mathcal{X} \cup \hat{\mathcal{X}} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the space of all natural images, $\hat{\mathcal{X}}$ is the space of all imperceptible adversarial images, and $\mathcal{Y} = \{0, 1\}$ is the space of ground truth binary labels, with 1 denoting an adversarial example. Since it is intractable to capture the entire space of natural and adversarial examples, we estimate $\mathcal{X} \cup \hat{\mathcal{X}}$ with the dataset $D = \bigcup_{i=1}^{N} \{(\mathbf{x_i}, 0), (\hat{\mathbf{x}}_\mathbf{i}, 1)\}$, where $\hat{\mathbf{x}}_i$ is an adversarial example obtained by attacking $\mathbf{x}$. An attack specific detector is obtained by restraining the threat model of adversarial examples included in D, and an attack agnostic detector seeks to approximate the space of all adversarial examples by sampling from multiple diverse threat models. The output of SimCatch on an input image $\mathbf{x}$ is

$$\text{SimCatch}(\mathbf{x}) := \mathbf{d}_{\phi,\boldsymbol{\omega}}(\mathbf{x}) = sgn(\mathbf{w}^\mathbf{T}\boldsymbol{\phi}(\mathbf{x}) + b) \tag{5.1}$$

where *sgn* is the sign function. Note that D does not need to consist of clean and attacked pairs; it can also be two unrelated sets of clean and attacked examples. We hypothesize that training on clean and correspondingly attacked pairs will lead to a more precise decision boundary, but we find in practice that using arbitrary clean samples also suffices.

In classification over k threat models, the vector weights $\mathbf{w}$ are replaced with a matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$.The bias b also now becomes a d dimensional vector $\mathbf{b}$. We refer to this model as SimClass with learnable parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}$ and denote it as $\mathbf{g}_{\boldsymbol{\phi},\boldsymbol{\theta}} : \bigcup_{i=1}^{k} \hat{\mathcal{X}}_{d_i} \rightarrow [k]$, where

$\hat{\mathcal{X}}_{d_i}$ is the space of adversarially perturbed images under a threat model defined by distance metric d_i. The training set $D = \bigcup_{i=1}^{k} \bigcup_{j=1}^{N} \{(\hat{x}_j^i, i)\}$ consists of N adversarially perturbed examples from each of the k threat models. The output of SimCat used as classifier on input x is then

$$\text{SimClass}(\mathbf{x}) := \mathbf{g}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) = \arg\max_{i \in [k]} (\mathbf{W}\boldsymbol{\phi}(\mathbf{x}) + \mathbf{b})_i \tag{5.2}$$

Both SimCat models are trained with a cross entropy loss and ℓ_2 regularization. Without loss of generality, we present the optimization formulation for SimCatch below.

$$\min_{\boldsymbol{\omega}} \sum_{(x,y) \in D} \mathcal{L}_{ce}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{x}), y) + \lambda \|\boldsymbol{\omega}\|^2 \tag{5.3}$$

Importantly, in training, the self-supervised encoder $\boldsymbol{\phi}$ is *fixed*. Thus, the number of learnable parameters scales linearly with the number of output classes and dimensionality of the embedding space of $\boldsymbol{\phi}$. Moreover, the optimization is now *convex*. Due to the low dimensionality of SimCLR's encoder and the convex nature of SimCat's optimization, the global optimum can be found efficiently, in both time and sample complexity. In our experiments, we set the regularization constant $\lambda = 1$, and use L-BGFS to obtain optimal parameters for SimCat's regularized logistic regression.

5.4.2 SimCat Variants

There are many additional modifications that can be made to SimCat to further improve its performance. The majority of the experiments do not use any variants, but in some cases we include the following:

- **Data Augmentation** is a common technique to improve generalizability of deep models. Naturally, data augmentation is very useful when there is limited data available. We utilize data augmentation to balance the dataset during adversarial training (algorithm 1). Our experiments with data augmentation show improvements in extremely low data settings, though only modest improvement in other cases.

- **Ensembling** can produce improved models by way of combining the outputs of multiple independently trained models. We ensemble detectors for specific poisoning types to improve the performance of our SimCatch based defense (table 5.3).

5.5 Detection and Classification

5.5.1 Evasion Attacks

5.5.1.1 Experimental Set up

We evaluate SimCat's detection and classification capabilities of evasion attacks from two datasets. We compare SimCat's performance to an analagous model that fits a linear layer atop fixed ResNet50 features learned via *supervised* ImageNet pretraining. The baseline differs with SimCat only in that it uses embeddings in a feature space learned with label supervision, highlighting how self-supervised features may better capture distinguishing nuances between the distributions of natural and adversarial images. We present results for a second baseline that additionally finetunes the pretrained ResNet50 in the supplemental material.

For SVHN [155], a dataset of street view house numbers with smaller images (32×32), we perform PGD ℓ_∞ and ℓ_2 attacks, with budgets of $\epsilon = 8/255$ and $\epsilon = 1.0$ respectively. We train a

SVHN		Training Samples per Attack				
Task	Attacks	2	5	10	25	50
Detection	PGD-ℓ_2	63.3(+8.8)	71.9(+11.5)	75.0(+11.9)	81.7(+13.1)	85.7(+12.0)
Detection	PGD-ℓ_∞	77.3(+15.0)	82.5(+13.9)	88.5(+14.0)	92.4 (+9.9)	94.2(+8.4)
Classification	PGD ℓ_2, PGD ℓ_∞	60.6(+8.9)	64.1(+12.8)	70.9(+16.3)	77.1(+21.0)	81.5(+19.6)
IMAGENET		Training Samples per Attack				
Task	Attacks	5	10	25	50	100
Detection	PGD-ℓ_2, PGD-ℓ_∞, PPGD, LPA, JPEG-ℓ_∞, StAdv, ReColor, CW-ℓ_2	68.5(+6.4)	71.5(+7.4)	74.3(+7.4)	76.5(+5.3)	79.2(+4.4)
Classification		27.1 (+7.7)	32.8(+10.2)	40.7(+11.8)	48.9(+12.8)	58.1(+15.3)

Table 5.1: Performance of SimCat for detection and classification using few training samples on SVHN (top) and ImageNet (bottom). In parenthesis, we denote the improvement gained by using SimCat compared to a baseline using supervised embeddings. For ImageNet, the detector is trained and evaluated over all eight attack types at once, and classification is done over all eight attack types.

detector for each PGD attack, and a classifier to distinguish between the two ℓ_p threats.

For ImageNet, we use the perceptual study data introduced in Section 3. Specifically, we take the attacks of the 'large' bound, which have a budget of $\epsilon = 8/255$ for the PGD-ℓ_∞ attack. The budget for each attack can be found in the original paper. Additionally, we perform Carlini-Wagner-ℓ_2 attacks on 200 other clean images. We train a *general* detector on all eight attack types, and also a classifier to distinguish between the eight attack types.

For each adversarial sample, we also have the original clean image. The samples are divided so that the pairs remain in the same set, ensuring that we never have a test image that is either the clean or adversarially perturbed version of a training image. Thus, the total training set size for a detection trial is equal to the number of training samples per attack times 2 times number of attacks, and times k for k-way classification.

In table 5.1, we present results averaged over ten trials, so to account for variability intro-

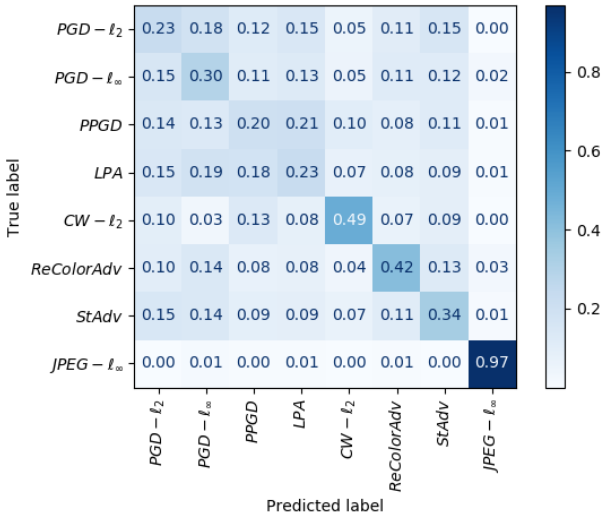


Figure 5.3: SimCat classification accuracy over eight diverse attack types. The classifier is fit on just 25 samples per class. Overall classification accuracy is 40.7%, an 11.8% increase over baseline.

duced by sampling such a small fraction of the data to train each model.

5.5.1.2 Results

SimCat outperforms the baseline across the board especially for SVHN, reaching increases in accuracy of as high as 21.0%. The efficiency of SimCat is highlighted in SVHN PGD-ℓ_∞ detection, where fitting a detector to just two adversarial examples yields 77.3% accuracy. SimCat's largest gains over the baseline comes in classification tasks (5.3), indicating that self-supervised features are more sensitive to the distinguishing characteristics of specific attack types.

Detection		Classification	
Task	Accuracy	Task	Accuracy
BP	85.3	Backdoor vs. Triggerless	68.9, 78.4*
CP	84.1		
General	64.5, 70.5*	5-way	52.4

Table 5.2: Results for SimCat detection and classification of poisoning attacks on STL10 using ten samples per poisoning. Asterisk indicates removing FC poisons.

5.5.2 Poisoning Attacks

5.5.2.1 Experimental Setup

We test SimCat on five poisoning attacks, including Bullseye Polytope (BP), Convex Polytope (CP), Feature Collision (FC), Clean-label Backdoor (CLBD), and Hidden-trigger Backdoor (HTBD). Poisons are generated using the white-box transfer learning set up as described in [156], where the attacker seeks to poison a fine-tuning set for transfer learning. This setting is generous to the attacker, as they only need to poison a linear layer appended to a fixed feature encoder, that they also have access to. Furthermore, the training set sizes are small. Specifically, the finetuning set only uses 2500 images, and the attacker can insert 1% (25) additional samples. We use the STL10 dataset, as an intermediary between SVHN and ImageNet. As an additional challenge, we use the SimCLR encoder as the fixed feature encoder – this means that the poisoning attacks directly cause collisions with clean samples in the same space we wish to use to distinguish them. We test how well clean target samples can be distinguished from poisons that (by design) would be in close proximity to the targets in SimCLR space. We also test SimCat as a poison defense.

Attack Type	Standard		SimCat		Ens. SimCat	
	Acc	PSR	Acc	PSR	Acc	PSR
BP	86	82	86	54	85	41.3
CP	86	24	86	11.3	85	4
FC	87	0	86	1.3	85	2
CLBD	87	0	86	0.7	85	0
HTBD	86	2	86	2	85	1.3
Avg	86	21.8	86	13.9	85	**9.7**

Table 5.3: Poison defense via SimCat detection. PSR is poison success rate. SimCat model is trained on CP and BP jointly. Ens. SimCat trains a CP detector and a BP detector separately, then filters any samples that are detected as poison by either detector. Both defenses only use *ten* samples of CP and BP poisons each.

5.5.2.2 Results

SimCat again shows strong detection and classification accuracy with high sample efficiency (5.2), particularly for BP and CP poisons, which happen to be the strongest. SimCat struggles with FC poisons, most likely since FC poisons are designed to directly collide with target representations, while BP and CP poisons surround a target instead. When excluding FC poisons, general detection rises to 70.5%.

We then apply SimCat detectors as a poison defense. We use BP and CP poisons to train the detectors, since those poisons are most lethal. Using ten samples each, we train an attack agnostic detector, and two separate detectors specific to each threat model, which are used as an ensemble detector, that only admits samples deemed clean by both detectors. Table 5.3 shows that both the general detector and the ensemble detector are effective, with the ensemble detector reducing poison success rate from 21.8% to 9.7%, while also maintaining high clean accuracy.

Trained on:	# of Samples	PGD-ℓ_2	PGD-ℓ_∞	PPGD	LPA	CW-ℓ_2	ReColor	StAdv	JPEG-ℓ_∞	Avg.
Single Attack	100	68.8	67.9	68.5	69.2	62.2	64.9	65.6	50.2	64.7
Union of Attacks	5	66.4	70.7	65.6	73.4	51.5	71.4	63.3	69.9	66.5
	20	71.1	76.6	69.0	80.1	51.1	74.6	66.1	79.4	71.1

Table 5.4: Generalization of SimCat detectors to unseen threat models. First row shows accuracy of detector trained on a single attack evaluated on all other attacks. Other rows contain accuracy of a SimCat detector trained on the union of all other attacks. The second column indicates the number of samples per threat model used in training.

5.6 Generalization to Unseen Models

Generalization of defenses to unseen attacks is of utmost importance because of the constant development of novel threats. In table 5.4, we see that SimCat generalizes surprisingly well given its simplicity. Even when trained only on a single threat model, some of the SimCat detectors achieve close to 70% detection accuracy on unseen attacks. The generalization of detectors trained on the union of attacks is also impressive, particularly given the sample efficiency. We observe that the detector trained on the union of attacks with just five samples per threat model (35 total) exceeds the average accuracy on unseen attacks achieved by detectors trained on a single threat model with 100 training samples.

In figure 5.4, we get a closer look at how each threat model generalize to others. The detectors trained on the perceptual attacks (PPGD, LPA) generalize the best to unseen threats. This invokes our motivating observation that the SimCLR embedding space seems to contain information pertinent to perceptibility. Understanding how human and machine perceptions differ is at the heart of many vision tasks, including adversarial robustness, and we encourage future work to further investigate the semantic meaning extracted in SimCLR and other self-supervised models.

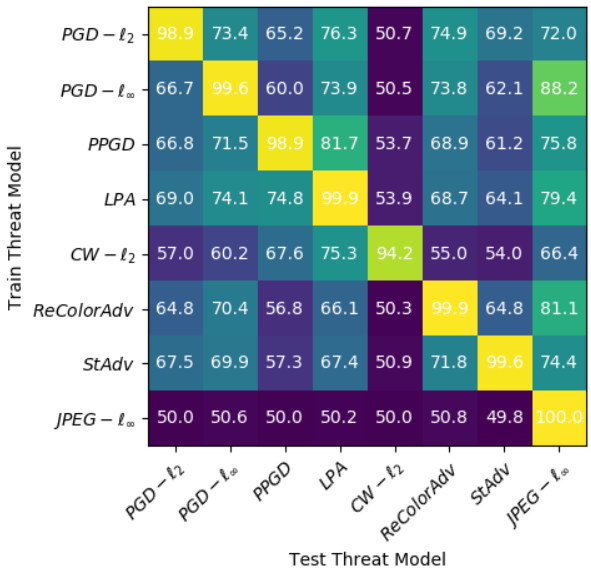


Figure 5.4: Generalizability of SimCat detectors to unseen threat models. Each detector is trained on 100 samples from a single threat model, and evaluated on all other models.

5.7 Hardening SimCat to an Adaptive Attack

In this section, we investigate the robustness of SimCat to an adaptive attack. An adaptive attack is an attack that is specifically crafted based on knowledge of a model's defense. By investigating adaptive attacks for SimCat, we can identify limitations of our model, and work towards mitigating them (i.e. via adversarial training) before the vulnerability is exposed and exploited by an adversary.

5.7.1 Attack Formulation

We consider a white box attack setting, where the attacker has knowledge of the base classifier *and* the SimCat detector. The ultimate goal is to cause a misclassification in the base classifier, but the adversary must first evade the SimCat detector. Denoting the base classifier as $\mathbf{f}$, the detector as $\mathbf{d}$, and an input-label pair as (x, y), we formulate the adaptive adversarial attack

problem as the following.

$$\boldsymbol{\delta} = \underset{\boldsymbol{\delta}, \|\boldsymbol{\delta}\| \leq \epsilon}{\arg\max} \mathcal{L}(\mathbf{f}(\hat{\mathbf{x}} + \boldsymbol{\delta}), y) + \mathcal{L}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\hat{\mathbf{x}} + \boldsymbol{\delta}), 1) \tag{5.4}$$

$\mathcal{L}$ denotes cross entropy loss. The detector outputs 1 for adversarial examples, so the adaptive attack seeks to flip this label by maximizing the loss incurred by it. We solve the above optimization with projected gradient descent, and find that the adaptive attack is somewhat effective against an undefended SimCat ImageNet detector, reducing accuracy by 30% (table 5.5).

5.7.2 Adversarial Training

We employ adversarial training (AT) to improve the robustness of SimCat to the adaptive attacks described in the previous section. Standard AT seeks to harden a network by crafting adversarial examples throughout training, and additionally training the model on the crafted examples with the original label. For SimCat, this amounts to the following min-max optimization, where $\mathbf{d}$ is the SimCat detector with parameters $\boldsymbol{\phi}, \boldsymbol{\omega}$, and $\mathbf{f}$ is the base classifier.

$$\min_{\boldsymbol{\omega}} \max_{\boldsymbol{\delta}; \|\boldsymbol{\delta}\| \leq \epsilon} \mathcal{L}_{ce}(\mathbf{f}(\hat{\mathbf{x}} + \boldsymbol{\delta}), y) + \lambda \mathcal{L}_{ce}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\hat{\mathbf{x}} + \boldsymbol{\delta}), 1) \tag{5.5}$$

SimCat AT is different from standard AT in a few ways. Standard AT usually takes a few steps of finding perturbations to increase the objective, followed by a few steps of updating model parameters to reduce the objective. In SimCat AT, the minimization step is solved to completion after having crafted adaptive adversarial examples, as opposed to only taking a few steps. This can be done efficiently due to the low number of parameters to solve for and the convexity of the

minimization problem. A couple other steps are needed for SimCat AT to be effective.

- Momentum updates are used to stabilize training. The importance of momentum updates is clear in figure 5.5, as the $\beta = \infty$ case (where the SimCat detector is replaced in each epoch with the optimal solution for detecting the current batch of adaptive adversarial examples) yields worse robustness than a standard SimCat.

- Along with the additional adaptive adversarial attacks, the original data is retained in the training set for each iteration, so to mitigate a robustness-accuracy tradeoff. To balance the dataset, an augmented copy of the clean samples is also added to the training set in each epoch.

Algorithm 1 Adversarial Training of SimCat: inputs are the base classifier $\mathbf{f}$ and data $\left(\{(x_i, \hat{x}_i)\}_{i=1}^N\right)$, where x is a clean sample and $\hat{x}$ is x after being adversarially perturbed.

Obtain initial parameters via standard SimCat:
$\boldsymbol{\omega} \leftarrow \textsc{FitSimCat}\left(\{(x_i, \hat{x}_i)\}_{i=1}^N\right)$
Augment clean data to obtain second copy:
$\{\tilde{x}_i\}_{i=1}^N \leftarrow \textsc{Augment}\left(\{x_i\}_{i=1}^N\right)$
for $t = 1, \ldots,$ # of epochs **do**
  Craft perturbations for adversarial examples to evade both detector and base classifier:
  $\{\delta_{i,t}\}_{i=1}^N = \textsc{AdaptivePGD}\left(\{\hat{x}_i\}_{i=1}^N, \mathbf{f}, \mathbf{d}_{\phi,\boldsymbol{\omega}}\right)$
  Solve SimCat with expanded dataset:
  $\boldsymbol{\omega}_t \leftarrow \textsc{FitSimCat}\left(\{(x_i, \hat{x}_i), (\tilde{x}_i, \hat{x}_i + \delta_{i,t})\}_{i=1}^N\right)$
  Apply momentum update to SimCat parameters:
  $\boldsymbol{\omega} \leftarrow (\boldsymbol{\omega} + \beta\boldsymbol{\omega}_t)/(1 + \beta)$
end for

5.7.3 Results

While the adaptive adversarial attack is somewhat effective (reducing attack detection accuracy by 34%), the SimCat AT algorithm completely recovers robust accuracy, while also improving overall accuracy. Table 5.5 shows the effect of AT and augmentation, which in tandem

Model	Accuracy	Robustness
SimCat	73.21	39.25
SimCat+Aug	74.87	37.40
SimCat+AT	74.23	67.95
SimCat+AT+Aug	**73.55**	**71.70**

Table 5.5: Ablation study on SimCat AT. Accuracy refers to attack agnostic ImageNet detection from Section 5.1. Robustness is measured as the percent of test adversarial samples that can be adaptively attacked with PGD-ℓ_2, $\epsilon = 2.0$ to be misdetected as clean. AT is done for 20 epochs. Aug refers to augmenting both the original clean and adversarial samples - this is distinct from AT+Aug, where only the clean samples are augmented to balance out the addition of adaptive adversarial attacks to the SimCat training set.

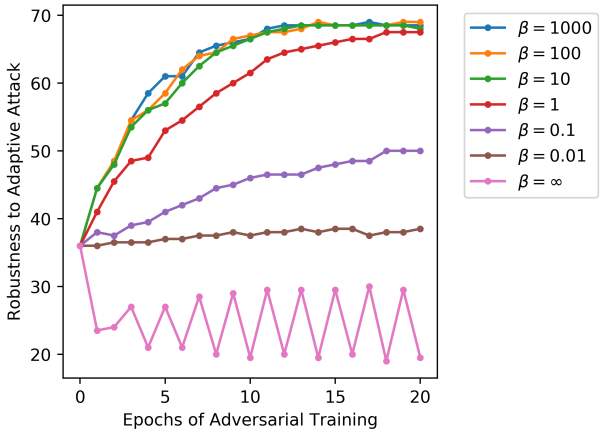


Figure 5.5: SimCat robustness to adaptive PGD-ℓ_2 over epochs of adversarial training with varied values of the hyperparameter β, which controls the momentum updates. Higher values of β lead to more emphasis on the linear classifiers solved in later epochs. Adversarial training is unstable without momentum ($\beta = \infty$).

become a very strong defense. Furthermore, the entire adversarial training procedure takes only about fifteen minutes, and all SimCat AT model results presented used only 25 training samples per attack. Thus, the SimCat framework lends itself to increased robustness via algorithm 1, without compromising training and data efficiency.

5.8 Conclusion

In this paper, we introduced SimCat, a sample efficient method for detection and classification of adversarial attacks. SimCat uses a linear model over embeddings of a pre-trained self-supervised model, SimCLR. SimCat is successful in detecting and classifying various types of adversarial attacks ranging from ℓ_p and non-ℓ_p evasion attacks to poisining attacks, likely because pre-trained SimCLR embeddings can be used to uniformly quantify perceptibility of various types of adversarial perturbations. Over various experiments on SVHN, ImageNet, and STL10 datasets, we demonstrate the effectiveness of SimCat using as few as two training samples per class. We have also studied adaptive attacks against SimCat and developed an adversarial training procedure that dramatically increases its robustness against such attacks while improving its clean accuracy.

Chapter 6: Text-To-Concept (and Back) via Cross-Model Alignment

This chapter details an ICML '23 paper co-authored with Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi [157]. I co-led this work with Keivan.

In this work, we observe that the mapping between an image's representation in one model to its representation in another can be learned surprisingly well with just a linear layer, even across diverse models. Building on this observation, we propose *text-to-concept*, where features from a fixed pretrained model are aligned linearly to the CLIP space, so that text embeddings from CLIP's text encoder become directly comparable to the aligned features. With text-to-concept, we convert fixed off-the-shelf vision encoders to surprisingly strong zero-shot classifiers for free, with accuracy at times even surpassing that of CLIP, despite being much smaller models and trained on a small fraction of the data compared to CLIP. We show other immediate use-cases of text-to-concept, like building concept bottleneck models with no concept supervision, diagnosing distribution shifts in terms of human concepts, and retrieving images satisfying a set of text-based constraints. Lastly, we demonstrate the feasibility of *concept-to-text*, where vectors in a model's feature space are decoded by first aligning to the CLIP before being fed to a GPT-based generative model. Our work suggests existing deep models, with presumably diverse architectures and training, represent input samples relatively similarly, and a two-way communication across model representation spaces and to humans (through language) is viable.

6.1 Introduction

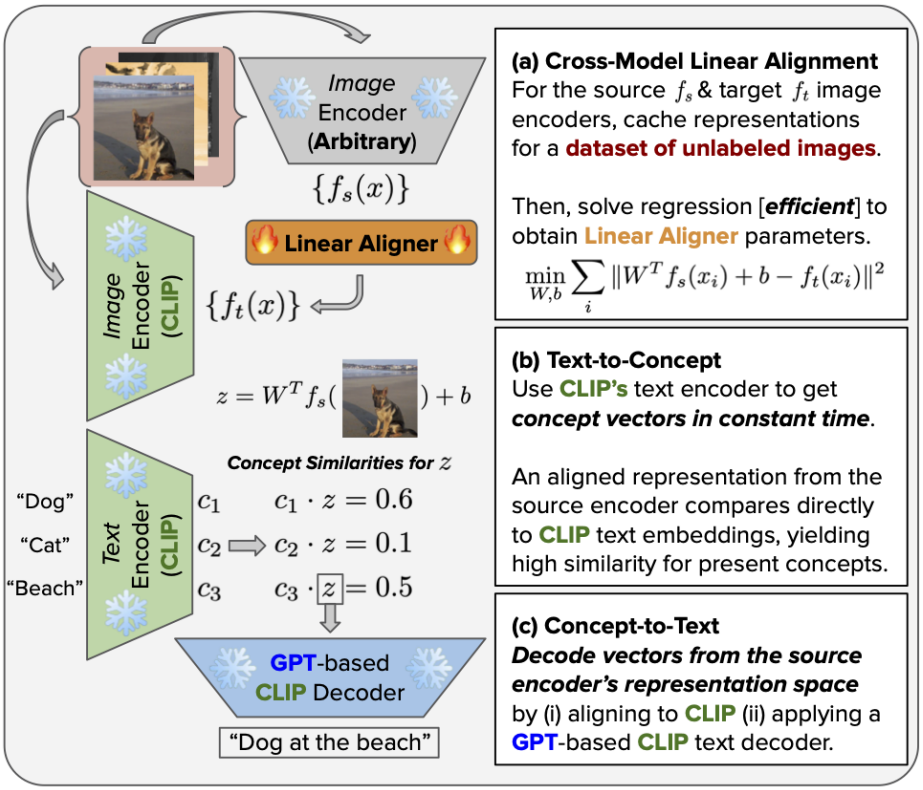


Figure 6.1: Overview. After *aligning* the representation space of a given image encoder to a CLIP image encoder, we can compare aligned representations of images to concept vectors obtained *directly from text* (typically, an example set of data is required to obtain each concept vector; our method is example-free, i.e. O(1) w.r.t. data collection). Using a GPT-based CLIP decoder, we can map arbitrary vectors in representation space to text. Our method yields efficient **interpretability**: we only train *one linear layer*.

The representation spaces of deep vision models are undoubtedly rich in semantic structure. However, these deep feature spaces are notoriously challenging for humans to interpret, mainly because it is hard for us to digest thousands of numbers at once. Unlike deep models, which encode concepts as vectors in high (e.g. $d = 2048$) dimensional spaces, humans have developed language to describe the world around us concisely. In this work, we propose a method to map text to concept vectors that can be compared directly to image representations obtained from

off-the-shelf vision encoders trained with *no text supervision*.

Our method works by *aligning* the representation space of a given vision model to the representation space of a CLIP [13] model. By design, the CLIP representation space is shared across jointly trained vision and text encoders. Thus, CLIP models already have text-to-concept built in, via the text encoder. To extend this capability to off-the-shelf models, we propose to learn a mapping between representation spaces. Specifically, we optimize a function to predict the representation of an image for a target model (i.e. CLIP) from the same image's representation for a source model (i.e. off-the-shelf vision model). We can then map the representations of the off-the-shelf model to CLIP space, where the aligned features would reside in the same space as the concept vector for the desired text.

The mapping function, however, may significantly change the semantics of its input. To prevent this, we restrict the hypothesis space of our mappings to be *affine transformations*. Despite their simplicity, we find that linear layers are surprisingly effective at performing feature space alignment, even between models with diverse architectures and training procedures. This observation suggests that despite drastically different approaches to training, diverse models seem to learn to store information in similar ways. Most notably, we can align model representations to CLIP, thus extending CLIP's text-to-concept abilities to existing models.

Figure 6.2 visually validates our approach: after encoding the concept "in a tree" in CLIP space and computing similarity with aligned representations from a self-supervised ResNet, the classes with the highest average similarity are reasonable, and images within them with the highest similarity prominently display the concept, while the least similar class instances do not. Stronger validation of our approach is found in performing **zero-shot classification using off-the-shelf encoders via text-to-concept.** Models achieve impressive zero-shot accuracy on many

tasks, often being competitive with a CLIP model that is larger, trained on many more samples with richer supervision, and most notably, directly optimized to align with the text encoder we use in text-to-concept. Surprisingly, zero-shot accuracy of off-the-shelf models surpasses the CLIP in a few cases, particularly for color recognition. While greatly expanding the use cases for existing models, these zero-shot abilities also support the belief that deep models learn many more abstract notions than what they are explicitly trained to know. Text-to-concept allows for uncovering and better utilizing the rich semantics hidden in existing models' representation spaces.

In addition to zero-shot learning for free, text-to-concept has several immediate interpretability applications, such as converting vision encoders to *Concept Bottleneck Models* (CBMs) [158] with *no concept supervision required*. CBMs decompose inference into a concept prediction step followed by class prediction using a white box model (i.e. linear head) on concept predictions, so that the contribution of each concept to the final logit can be precisely computed. Typically, CBMs require concept supervision in addition to class labels, but with text-to-concept, we can replace concept predictions with concept similarities, obtained by comparing aligned representations to the vector obtained for the desired notion. Then, CBM training reduces to simply training a linear layer to predict class labels from pre-computed similarities of aligned image representations to a set of concept vectors. We illustrate this application on RIVAL10 data [15], which has attribute labels, though we only use these labels to verify our zero-shot concept prediction approach. Indeed, we obtain an AUROC of 0.8 for RIVAL10 attribute prediction in the zero-shot manner described, leading to a highly accurate (93.8%) resultant CBM with desired interpretability benefits (see Figure 6.7).

Next, we show text-to-concept can demystify large datasets, as the distribution of similarities between a bank of text-to-concept vectors and aligned representations of the data essentially

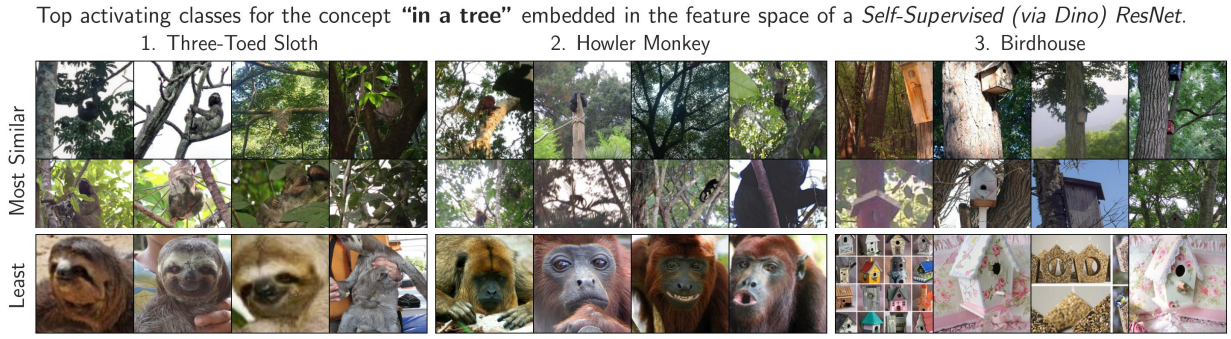


Figure 6.2: Qualitative validation of text-to-concept. ImageNet classes are sorted by the average cosine similarity of the CLIP embedding for "in a tree" to the *linearly aligned* Dino ResNet representations of images within each class. Highest ranked classes indeed often appear in a tree, as is evident by the most similar instances. The least similar instances appropriately do not contain the concept.

summarizes what concepts are present, explaining the data distribution in human terms. This can be applied to diagnosing distribution shifts, as we can inspect the shift w.r.t. to human-understandable concept similarities. For example, when comparing ImageNet to ObjectNet [37], we can show that the distribution of similarities for the "indoors" concept shifts dramatically, capturing the essence of why ObjectNet poses a challenge: images in ObjectNet were taken in people's homes. Another way text-to-concept aids in engaging with large datasets is via concept-based image retrieval. Using *concept logic*, we query the image representations for a given model that satisfy a set of concept similarity thresholds, allowing for greater human control of the importance of each concept in the search, yielding reasonable results in finding specific images in a large corpus.

Lastly, we close the human-machine communication loop by introducing *concept-to-text* to directly decode vectors in a model's representation space. Our implementation aligns the model's space to CLIP, and then leverages an existing CLIP space decoder [159] that uses a CLIP embedding to guide the output of GPT-2. The existing decoder was intended for image captioning, though we demonstrate that its abilities extend to general vectors (i.e. not obtained

from a single image) from non-CLIP models after alignment. Specifically, we decode the vectors in the classification head of three ImageNet trained models. We then use a human study to verify that the decoded captions describe the class associated with each vector, finding that our simple method works in over 92% of cases.

Our methods extend the capabilities of multi-modal models like CLIP to other models that are trained on much smaller uni-modal datasets and with weaker supervisions. This can be useful when a model more accurate or smaller than CLIP for a specific domain is desired, or when training CLIP-like models is infeasible, due to the large corpus of image-caption pairs needed. Moreover, since our approach can be applied to interpret any model's representation space, while only requiring the training of a linear layer, its potential impact is very high, as text-to-concept is easy to plug-in and has a breadth of applications. The implications of our work are also startling: first, the success of linear representation space alignment indicates that diverse models ultimately represents inputs relatively similarly. The emergent zero-shot abilities of existing models suggests an under-utilization of models we already have. Finally, the synergy we display between CLIP, GPT, and existing models, coupled with the ability to communicate across these models and back and forth with humans, makes the prospect of diverse models collaborating with minimal tuning very promising.

6.2 Review of Literature

Our alignment of model representation spaces is related to stitching, first introduced by Lenc and Vedaldi [160], who train linear layers to merge top and bottom chunks of different models, resulting in "franken-CNNs". Stitching was revisited by Bansal et al. [161], who aimed

to showcase how it can be used as a tool to quantify the quality of representations towards learning how to obtain better representations, and Csiszárik et al. [162], who consider different ways to train stitching layers. We note these works typically stitch together models of the same architecture, where as we consider a much more diverse set of models. Also, those works focus on comparing representations to one another, while we aim to relate representations to human notions.

Namely, we seek to obtain concept vectors within the representation space of off-the-shelf models from text. Also known as concept activation vectors (CAVs), Kim et al. [163] popularized the study of directions corresponding to human concepts in deep feature spaces, as well as the sensitivity of model outputs to changes along these directions, so to interpret deep networks. One limitation is the necessity of example sets of data to define CAVs. More recent efforts automatically discover CAVs [164, 165, 166], though annotating the discovered concepts with language is not straightforward, which motivates our concept-to-text method.

The rise of joint vision-language models like CLIP [13] make it possible to interpret vision space with text, as well as perform zero-shot classification. Follow up works leverage CLIP to annotate neural nodes [167] or to distill failure modes [168] of non-CLIP models. However, they engage probe datasets or exemplars to communicate with CLIP space, while our method directly aligns representation spaces. Zhai et al. [169] show that a language model can be tuned to a fixed image encoder to enable zero-shot image retrieval from text (or vice versa), and Merullo et al. [170] demonstrate that a linear mapping suffices in some cases, though both works require large image-text datasets in their training procedures, while ours only uses unlabeled image data. Recently, Moschella et al. [171] devise a zero-shot method for communication across representation spaces based on relative positions to anchor points, which supports our claim that representation

spaces are sufficiently equivalent to where the extension of CLIP's inherent text-to-concept ability to off-the-shelf models via linear alignment is viable.

6.3 Model Alignment

We use $\mathcal{X}$ to denote the set of all possible input images. Let $D_{\text{train}}, D_{\text{test}} \subset \mathcal{X}$ denote the training and test datasets. We define a *vision encoder* as a model f that maps images $x \in \mathcal{X}$ to vectors $f(x) \in \mathbb{R}^d$. Given two vision encoders f_s, f_t, *representation space alignment* of model f_s to model f_t is the task of learning a mapping $h : f_s(\mathcal{X}) \to f_t(\mathcal{X})$. We restrict h to the class of affine transformations, i.e., $h_{W,b}(z) := W^T z + b$.

To maximally retain the original semantics of representation spaces, we design the following optimization problem

$$W, b = \arg\min_{W,b} \frac{1}{|D_{\text{train}}|} \sum_{x \in D_{\text{train}}} \|W^T f_s(x) + b - f_t(x)\|_2^2.$$

The above optimization can be viewed as multiple linear regression problems; thus we evaluate the linear alignment on D_{test} by considering the quality of the solution on those linear regression problems. We use *Coefficient of Determination*, i.e., R^2 which is the proportion of the variation in the dependent variables that is predictable from the independent variables. Furthermore, we note that for the vision encoder f_s, there usually exists a *classification head* $g_s : \mathbb{R}^d \to \mathcal{C}$ that classifies a representation in the space of model f_s. Indeed, the predicted label for input x is $g_s\left(f_s(x)\right)$. Note that $\mathcal{C}$ denotes the set of labels, e.g., ImageNet classes. We define *aligned accuracy* as the accuracy of classification on D_{test} when we use f_s as the vision encoder, then do the linear transformation to obtain the corresponding representation in space of f_t, and

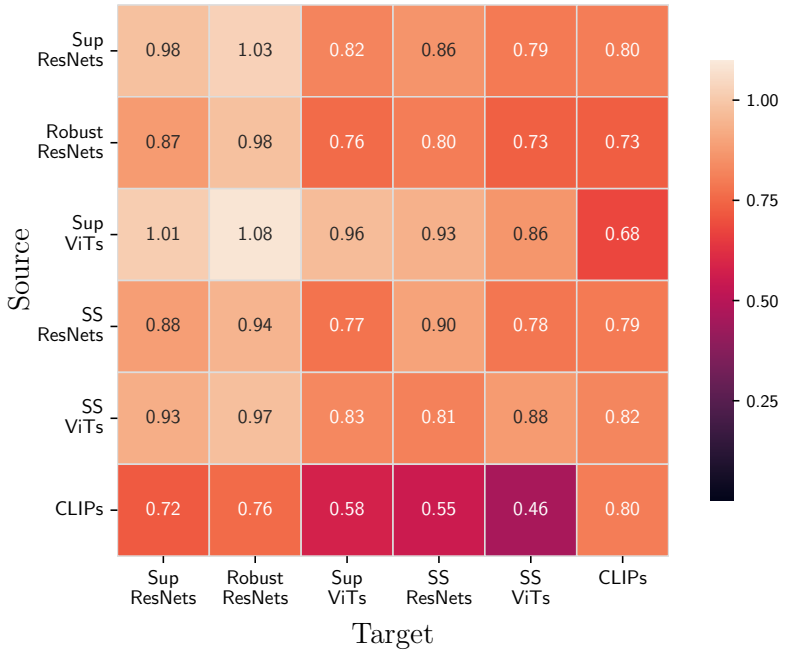


Figure 6.3: Heatmap of retained accuracy. The value in row r and column c is the average of retained accuracy when doing alignment from all models in group r to all models in group c. Note that "Sup" refers to supervised while "SS" refers to self-supervised models. All models except CLIPs are trained on ImageNet.

finally, use g_t for classification. If alignment works well, aligned accuracy should be admissible and comparable to the accuracy of model f_t when no alignment is used. Then, we define *retained accuracy* as the ratio of aligned accuracy to the accuracy of model f_t without any alignment. Note that we use ImageNet-1K train and test datasets as D_{train} and D_{test} in linear alignment.

Interestingly, we observe that simple linear alignment works well in terms of both R^2, and aligned accuracy across various models. Figure 6.3 shows the aligned accuracy between diverse pairs of models. In the scope of linear alignment, we further consider the sample efficiency of optimizing linear alignment as well as investigating linear alignment in space of top principal components of representation spaces, where we roughly see strong, near identity correspondence between the top principal components of different models.

According to Figure 6.3, various models are highly alignable to CLIP models. This is sur-

prising as CLIP models are trained on other datasets than ImageNet and their training procedure involves vision/text supervision which is drastically different from other models. High-quality alignment to CLIP representation space enables models to adopt a wide variety of CLIP models capabilities, which we analyze in this work. On the other hand, we observe that retained accuracy when aligning CLIP models to other models is not high. This is mainly due to the fact that CLIP models encode images and texts in relatively low-dimensional spaces and in linear regression, approximating dependent variables becomes harder as the number of independent variables decreases. Indeed, linear alignment works worse when we align from a representation space with lower dimensionality to a representation space with higher dimensionality.

We note that linear alignment can also be optimized in other ways. Namely, when a specific classification task is of interest, one may optimize the aligner directly for that task. That is, one can minimize $\ell(g_t(W^T f_s(x) + b), y)$ over a set of image-label pairs $\{(x, y)\}$, where ℓ is the cross entropy loss. One can replace labels y with predictions $g_s(f_s(x))$ if remaining faithful to the source model is desirable; this also alleviates the need for labeled data to train the aligner. Lastly, a data-free approach is to optimize the aligner so that g_s becomes equivalent to the composition of the aligner and g_t. Essentially, here we seek to re-express the classification head of the source model as a function of the learnable aligner and the fixed classification head of the target model.

6.4 Text to Concept

Leveraging representation space alignment, specifically to CLIP, we perform text-to-concept, where text descriptions of semantic concepts are encoded as vectors that can be directly compared (i.e. via cosine similarity) with the aligned features of images obtained from an off-the-shelf

vision encoder (see Figure 6.1). Despite its simplicity, alignment-based text-to-concept is surprisingly effective, which, after further detailing our method, we demonstrate qualitatively and quantitatively in this section. Notably, we show that the similarities of aligned image representations to class vectors obtained via text-to-concept enables *zero-shot classification for non-CLIP models off-the-shelf*, with zero-shot accuracy of much simpler models at times exceeding CLIP's.

6.4.1 Method Details

We define text-to-concept as a procedure for obtaining vectors corresponding to concepts described as text that can be directly compared (i.e. via cosine similarity) to image representations from a fixed vision encoder. Our method begins with a string describing some concept, like "red food". We then prepend this string with a number of template prompts (e.g. "a photo of {}"); we use the same template prompts as in CLIP's original paper for ImageNet zero-shot classification. Then, we embed the templated text to CLIP space using CLIP's text encoder, and average the resultant vectors over all templates to obtain a single concept vector (as is standard). For some object agnostic concepts, such as contexts like "in a tree", we can encode a general prompt like "a photo of an object in a tree", or we can obtain a more refined vector by encoding "{*prompt*} {*class name*} in a tree", averaging over all choices for *class name* and *prompt*. There are countless ways to prompt engineer; we elect to use general prompts in most cases, as prompt engineering is not the focus of our work.

Then, for a given model, we train a linear layer to align its representation space to CLIP; specifically, we use CLIP ViT-B/16. We pass ImageNet training images to the given model's feature encoder and CLIP's vision encoder, resulting in a dataset of paired representations with

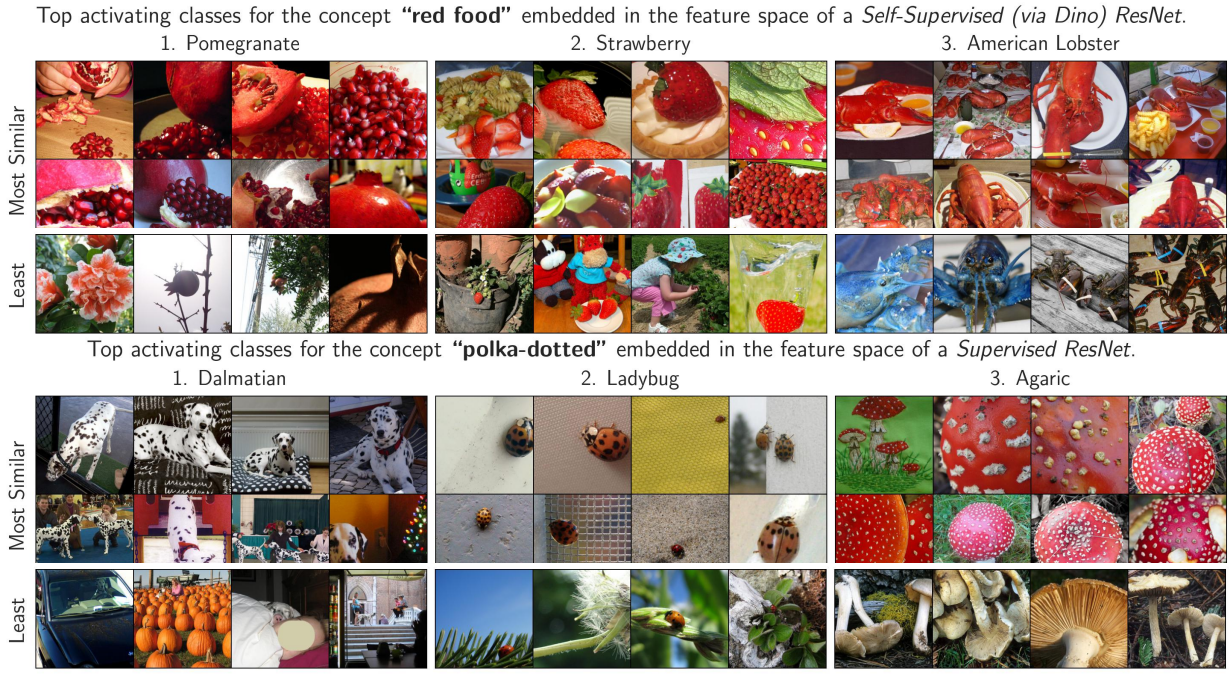


Figure 6.4: Text-to-concept can encode finer-grain concepts, like combinations of concepts (*"red food"*) or textures (*"polka-dotted"*).

which we train our aligner (Section 6.3). Now, we have two functions that map to CLIP's vision space: the CLIP text encoder (since the text and vision representation spaces are shared), and the composition of the given model's encoder with the linear aligner. Since the concept vector obtained via CLIP's text encoder and aligned representations from the given model are both mapped to the same space, we can compare them directly, thus satisfying our definition of text-to-concept. Alternatively, we could train an aligner from CLIP to the given model's representation space, and align the text embedding instead of the features. We found this method to be less effective, possibly because the dimensionality of CLIP space is lower than most models we study. Since our aligner is a simple affine transformation, alignment minimally changes the content of the representation obtained from the off-the-shelf model. Also, note the efficiency of our approach: after training a linear layer once, we can encode any number of new concepts from text at no additional training cost.

Qualitative Validation: Figures 6.2 and 6.4 show images selected based on the cosine similarity of their aligned representations (obtained using off-the-shelf encoders and trained linear aligners) to certain concept vectors. For each concept, we present the classes with the highest average similarity, as well as the most and least similar images within them. The retrieved classes are sensible for each concept (e.g. *American Lobster* for "red food"). Sorting images within each class separates examples where the concept is extremely prominent from those where the concept is absent (e.g. images of uncooked lobsters are least similar to the "red food" concept). Note that the models used to obtain the image representations differ in architecture, training objective and supervision from CLIP, and most notably, they have not been trained with any text/concept supervisions. Thus, it is surprising that we can easily connect these visual concept representations to the CLIP text embeddings.

Nonetheless, over a range of concept types (a context, a combination of color and a concept, and a texture), our visualizations qualitatively validate our proposed text-to-concept approach. We now turn to zero-shot classification for quantitative validation.

6.4.2 Zero-Shot Classification

CLIP models perform zero-shot classification by comparing image representations to embeddings of text strings describing each class: the predicted class is the one whose text embedding is most similar to the test image's representation. This is referred to as zero-shot since no labeled instances from the candidate classes are used. Considering classes as concepts, we can then use text-to-concept to obtain vectors that are directly comparable to aligned representations from off-the-shelf vision encoders, thus extending CLIP's zero-shot capabilities. The accuracy of

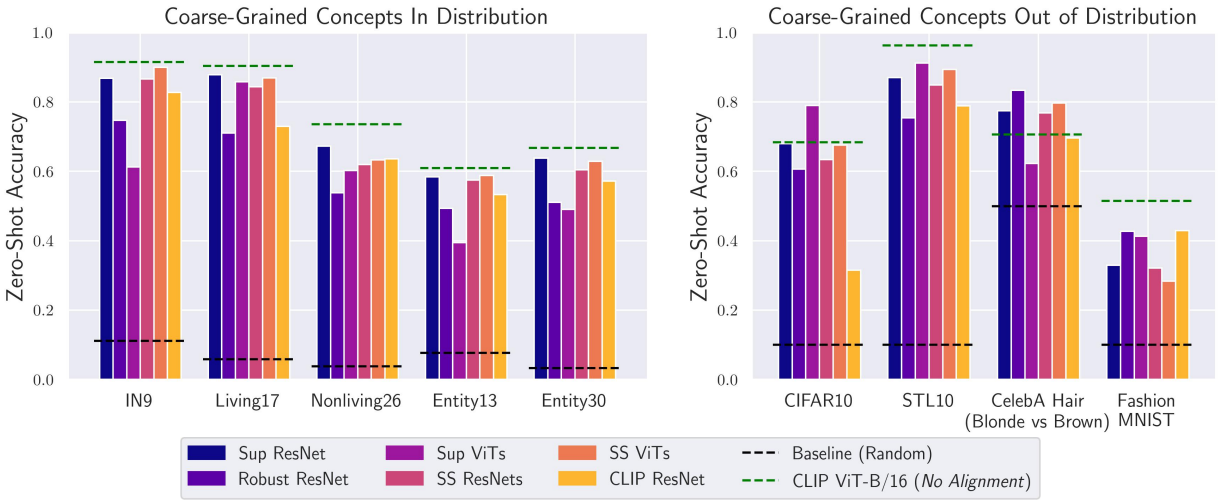


Figure 6.5: The zero-shot capabilities of CLIP can extend to off-the-shelf vision encoders via alignment based text-to-concept. (**Left**) Models trained on ImageNet can recognize coarse categorizations of ImageNet classes, despite never explicitly being taught them. (**Right**) Off-the-shelf models remain strong zero-shot classifiers even when images are out of distribution. In some cases, they surprisingly surpass the accuracy of the CLIP vision encoder whose jointly-trained text encoder was used to embed each class vector.

zero-shot classification serves as a quantitative measure of the quality of text-to-concept vectors. Indeed, when concept vectors align better with representations of samples in the class, zero-shot accuracy is higher. Thus, we explore zero-shot classification over many datasets to shed insight on when and how well text-to-concept works. We consider models over diverse architectures and training procedures, though all models are roughly equal in size ($\sim$25M parameters) and are only trained on ImageNet (except for CLIP). Also, the baseline CLIP model (ViT-B/16) whose text encoder is used to embed concepts is much larger in size ($\sim$80M parameters); this baseline is intended more so as an upper bound.

First, we ask if models can recognize new categorizations of the data they were trained over. Namely, we consider coarse grained categorizations of ImageNet classes (e.g. distinguishing *insects* from *carnivores*, see [39, 172]). We also investigate if these coarse grained concepts can still be recognized as image data is taken out of distribution. Figure 6.5 displays the results. We

observe impressive zero-shot performance in both cases. For example, on a 17-way classification problem, self-supervised ViTs achieve 85% accuracy, despite never receiving supervision about these classes, or any classes at that. Shockingly, in a few cases, even the performance of the CLIP model whose text encoder (with which it was jointly trained) was used to obtain concept vectors is surpassed.

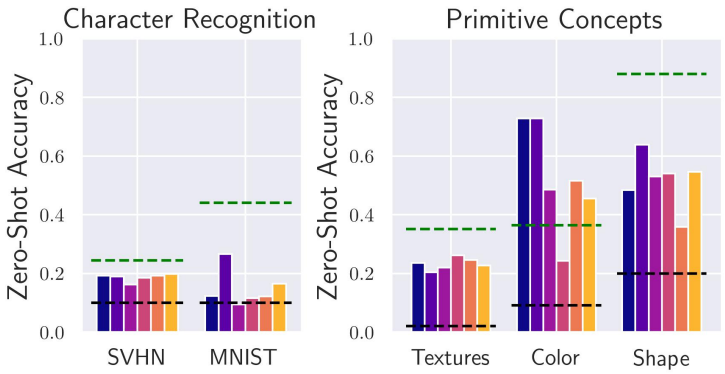


Figure 6.6: Edge cases for zero-shot classification. (**Left**) Models struggle with OCR. (**Right**) Models can recognize some primitive concepts by name. Same legend as figure 6.5.

To stress test text-to-concept, we consider tasks that require models to recognize characters (specifically digits) or primitive concepts, like textures, colors, and shapes. We observe most models only marginally surpass random accuracy for character recognition tasks. Oddly, the adversarially trained ResNet is roughly twice as good as other models in zero-shot MNIST classification, though it still performs far worse than the baseline CLIP model, which also struggles. This suggests models simply may not have any notion as to what distinguishes digits from one another, which is not surprising given that it would not be very useful for understanding ImageNet images. On the other hand, models achieve far better than random performance in recognizing primitive concepts, which appear in ImageNet as low level features for more abstract notions. Interestingly, color recognition is a task where most models outperform the CLIP baseline, sug-

gesting the CLIP ViT may have reduced color sensitivity relative to other primitive concepts.

While these experiments validate our proposed text-to-concept method, it is also remarkable that these off-the-shelf models, who have much smaller training sets (roughly 0.3% the size of CLIP's) and receive far less supervision, are comparable to CLIP in recognizing the unseen classes we consider. This suggests that models learn far more than what they are taught. In other words, models discover many semantic concepts and organize their representation spaces so that these concepts are roughly linearly separable, even when they are only explicitly directed to separate 1000 classes or to simply draw representations of similar inputs close to another. Thus, even models trained with elementary techniques likely contain much richer representation spaces than their use case requires. The success of transfer learning supports this claim, as a small amount of labeled data is sufficient for a model to recognize 'new' concepts, implying they had some notion of the concepts before. Text-to-concept can enable better understanding and utilization of these rich representation spaces, *without requiring new labeled samples*.

6.5 Additional Applications of Text-to-Concept

6.5.1 Concept-Bottleneck Networks for Free

The zero-shot results suggest models are already aware of many concepts beyond those which they are directly trained to learn. One case where knowledge of concepts related to the classification task is salient is Concept Bottleneck Models (CBMs) [158]. CBMs are intepretable by design, as they first predict the presence of concepts using a black box, and then obtain class logits with a white box (e.g. linear layer) atop concept predictions. Thus, the contribution of each concept to the predicted logit can be computed directly, allowing predictions to be faithfully

explained with semantic reasons. A major barrier to using CBMs is that they require concept supervision, which can be prohibitively expensive. Text-to-concept, however, alleviates this constraint, thanks to zero-shot concept prediction.

We use RIVAL10 classification [15] as an example for how a CBM can be implemented with *no concept supervision* using text-to-concept. RIVAL10 is an attributed dataset, though we do not use these labels during training. We use RIVAL10 because a linear classifier operating on ground truth attribute labels achieves 94.5%, indicating that a CBM could be effective. Further, the attribute labels allow for quantifying the quality of the zero-shot concept vectors we obtain.

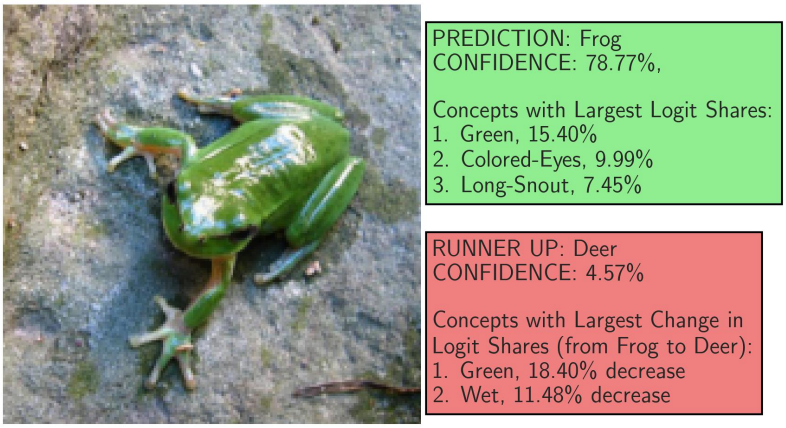


Figure 6.7: Example inference for a Concept Bottleneck Model (CBM) obtained via training a linear layer on zero-shot concepts. Since logits in the CBM are linear functions of concept scores, we can precisely quantify the contribution of concepts to each logit.

To implement the network, we use text-to-concept to encode the 28 attributes annotated in RIVAL10 as vectors in CLIP space. We then compute the similarities between the attribute vectors and aligned (to CLIP) features from an ImageNet pretrained ResNet-50. Finally, we fit a linear layer atop image-attribute similarities (i.e. in representation space) to predict class labels. Note that the only training we conduct is that of the final classification head and of the aligner, both of which are linear layers, making them time and sample efficient to optimize. The

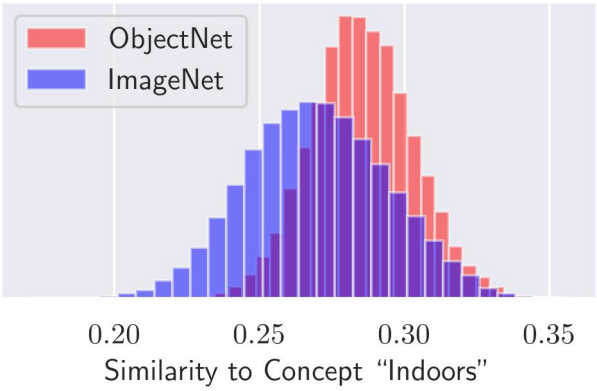


Figure 6.8: Concept similarities can reveal distribution shifts, like in ObjectNet, where photos are taken within people's homes.

resultant CBM achieves 93.8% accuracy, and yields the desired interpretability advantages, as shown in figure 6.7. Moreover, using image-attribute similarities (via text-to-concept) as a score for predicting attributes achieves an AUROC of 0.8, with 72% of attributes achieving at least an AUROC of 0.75. Thus, zero-shots concepts are relatively accurate in predicting RIVAL10 attributes.

6.5.2 Concept-Based Dataset Summarization and Distribution Shift Diagnosis

The interpretability benefits of text-to-concept also apply to demystifying large datasets. Specifically, one can discern the presence of a concept in their data by using text-to-concept to obtain a corresponding vector, and computing the similarity of this vector to all aligned images representations. As modern datasets continue to grow, the need for efficient concept-based summaries of these datasets will also grow; text-to-concept can provide such summaries easily.

Moreoever, one can track the distribution of concept similarities for a stream of data over time. Suppose for example a model is deployed to a new setting and it begins to fail. By comparing the distribution of concept similarities in the training set to the new data, one can diagnose

the distribution shifts at play. As a proof of concept, we inspect ObjectNet [37], a challenging distribution shift for ImageNet models consisting of images taken within people's homes. Figure 6.8 shows the distribution of similarities between the vector for the concept 'indoors' and aligned image representations obtained from a ResNet-50 of ImageNet and ObjectNet samples. For ObjectNet, the distribution is significantly (as determined with a Kolmogorov-Smirnov test) shifted to the right compared to ImageNet. In practice, one may maintain a bank of concepts and track similarities over their stream of data, automatically flagging concepts which experience significant shift.

6.5.3 Concept Logic for Image Retrieval

Text-to-concept enables the computation of the similarity of a model's representation for an image to an arbitrary concept. Given a corpus of data and a vision encoder trained to represent said data, we can retrieve images using text, based on their similarity to text-to-concept vectors. While one may combine all keywords in a string to obtain a single composite concept vector, we observe suboptimal performance with this approach, as words receive imbalanced attention and negations are often ignored by CLIP's text encoder.

A simple alternative is to retrieve images that satisfy a set of conditions. For example, instead of searching for "a dog on the beach at sunset", we can separately encode the concepts 'dog', 'on the beach', and 'at sunset'. We then filter images based on their similarity for each concept; we use thresholds based on the distribution of similarities for a given concept (i.e. at least 3 standard deviations above the mean). Analogously, we can encode negative conditions by requiring similarity to be below some threshold. Figure 6.10 presents our method as a diagram .



Figure 6.9: Images retrieved based on similarity of their representation to multiple text-to-concept vectors. Retrieved images satisfy the multiple conditions listed above each image. $\sim$ denotes 'not'. Concept logic with text-to-concept enables searching over images, while allowing the use of any vision encoder to represent them.

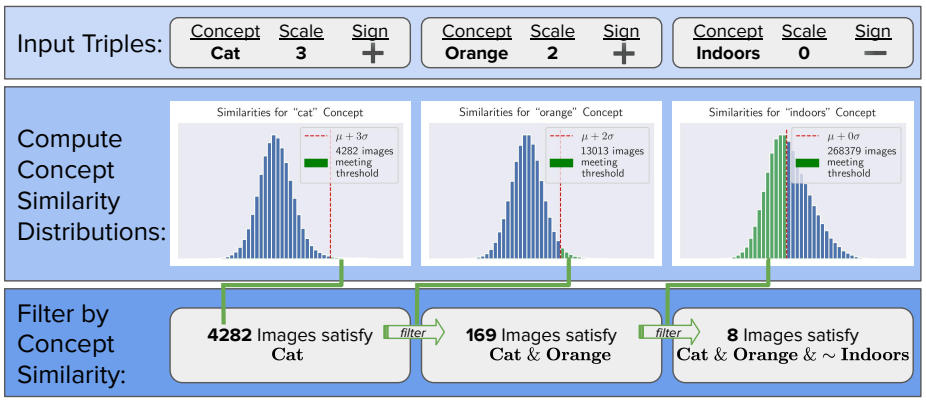


Figure 6.10: Overview of our concept logic image retrieval technique. Instead of directly searching for images of *orange cats that are not indoors*, we decompose the query into concept constraints of varying degree. Our method grants the querier more control in their search, and also circumvents limitations of CLIP's text encoder when processing negations and longer queries.

We demonstrate the effectiveness of concept logic by retrieving rare images (visualized in Figure 6.9) from a set of keywords. While concept logic for image search can be done over CLIP vision embeddings, the results may be suboptimal when querying over a specific dataset for which CLIP was not finetuned, particularly compared to a vision model trained on that dataset.

Swin (S)	ResNet-50	Dino ViTs8
94.48%	95.14%	92.18%

Table 6.1: Percent of captions for decoded class vectors deemed relevant to images in the class by human annotators.

6.6 Concept-to-Text

Text-to-concept grants insight into the representation spaces of deep models by mapping semantic notions expressed as words directly to concept vectors. However, humans still need to conjecture what concepts may be relevant before probing a representation space. We now ask, can we directly map concept activation vectors to text? We refer to this as *concept-to-text*, and propose an implementation using alignment to CLIP and generative language models (Figure 6.1).

Similar to how we that observe diverse vision models learn to store information in similar ways, allowing for cross-model alignment, we argue that language models and vision models similarly learn much of the same information, and can thus be plugged into one another with ease. Recent work supports this claim, as vision models have been stitched to generative language decoders to perform image captioning and visual question answering [170, 173]. Notably, Mokady et al. [174] captions an image by feeding its CLIP image embedding through a finetuned version of GPT-2 [175]. A follow up work, ZeroCap [159], similarly decodes with GPT-2 while receiving guidance from a CLIP embedding, but does so without requiring any tuning of either CLIP or GPT-2. We elect this method as it further demonstrates how existing models can work together off-the-shelf. However, other decoders could easily be put in place of ZeroCap if desired. We highlight this flexibility as it entails that concept-to-text will continue to improve as individual components are improved (e.g. GPT-2 $\rightarrow$ Chat-GPT).

With our linear aligners, we can already map representations from off-the-shelf encoders to CLIP. Thus, with no additional training, we can perform elementary concept-to-text by simply feeding aligned features to ZeroCap. While ZeroCap expects CLIP embeddings of natural images as input, we conjecture that passing a vector encoding some semantic notion can similarly be decoded. To asses this claim, we consider the task of decoding classification head vectors. These vectors exist in the original model space, and should encode information relevant to their corresponding ImageNet class. Thus, we can quantify the effectiveness of our elementary concept-to-text method by seeing if decoded class vectors indeed describe the desired class. Specifically, we use the prompt "Image of a " and set the desired sequence length to 1 so that a single word is decoded per class vector. We perform a human study to answer this question, showing MTurk workers a collage of images from a given class, along with the caption obtained from decoding the class vector, and asking if this caption is relevant to the images shown. The results, shown in Table 6.1, show that in over 92% of cases, our naive approach to concept-to-text appears effective in decoding class vectors for three diverse models.

6.7 Conclusion

In this work, we demonstrate the feasibility of training a single linear layer to map the representation spaces of diverse vision encoders, suggesting that these models represent images surprisingly similarly. We then utilize this method to align arbitrary vision encoders to CLIP space, essentially extending the multi-modal capabilities of CLIP to many new models trained with far less constraints (e.g. paired image-text supervision, large dataset size, contrastive learning). Our method unlocks new powers in existing models, and opens the door for better data and

model concept-based interpretability, without requiring costly concept annotations. We hope our work inspires others to explore new ways in which existing models can be further utilized with minimal additional compute, via allowing these models to work together.

Chapter 7: Unearthing Skill-Level Insights for Understanding Trade-Offs of Foundation Models

We now turn to a new flavor of interpretability to understand a more complex suite of models. Namely, we seek to obtain richer evaluation signal on so-called foundation models, which are intended to be far more general purpose than the image classifiers we studied in Part 2. This chapter details a paper under submission to ICLR '25, co-authored with Varun Chandrasekaran, Vidhisha Balachandran, Safoora Yousefi, Thomas Fel, Soheil Feizi, Besmira Nushi, Neel Joshi, and Vibhav Vineet [176].

With models getting stronger, evaluations have grown more complex, testing multiple skills in one benchmark and even in the same instance at once. However, skill-wise performance is obscured when inspecting aggregate accuracy, under-utilizing the rich signal modern benchmarks contain. We propose an automatic approach to recover the underlying skills relevant for any evaluation instance, by way of inspecting model-generated *rationales*. After validating the relevance of rationale-parsed skills and inferring skills for 46k instances over 12 benchmarks, we observe many skills to be common across benchmarks, resulting in the curation of hundreds of *skill-slices* (i.e. sets of instances testing a common skill). Inspecting accuracy over these slices yields novel insights on model trade-offs: e.g., compared to GPT-4o and Claude 3.5 Sonnet, on average, Gemini 1.5 Pro is 18% more accurate in *computing molar mass*, but 19% less accurate in *applying*

constitutional law, despite the overall accuracies of the three models differing by a mere 0.4%. Furthermore, we demonstrate the practical utility of our approach by showing that insights derived from skill slice analysis can generalize to held-out instances: when routing each instance to the model strongest on the relevant skills, we see a 3% accuracy improvement over our 12 dataset corpus. Our skill-slices and framework open a new avenue in evaluation, leveraging skill-specific analyses to unlock a more granular and actionable understanding of model capabilities.

7.1 Introduction

Recent years have seen benchmarks evolve to keep up with ever-advancing models. While classical benchmarks tested specific capabilities, like recognizing digits [177] or classifying sentiment [178], modern benchmarks measure proficiency in numerous capabilities simultaneously, drawing questions of increasing difficulty from more diverse domains [179, 180, 181]. As the questions we test models on have grown more complex, aggregate performance measures provide less understanding about model proficiency in specific abilites. For example, as shown in Figure 7.1, we find that over a dozen benchmarks, GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet [182, 183, 184] achieve overall accuracies within 0.4% of one another, leaving an open question: Are these models all the same, or are valuable insights being averaged away?

Manual annotations of categories across instances (e.g. [185] and [186] include 'ability' tags) enable going beyond accuracy, but at a cost for benchmark creators that rises with the number and difficulty of test questions. This results in few (if any) and non-standardized annotations, restricting cross-benchmark aggregation, even though many benchmarks have large overlap in the (implicit) skills they test [187, 188]. Prior works show promise in automatically grouping inputs

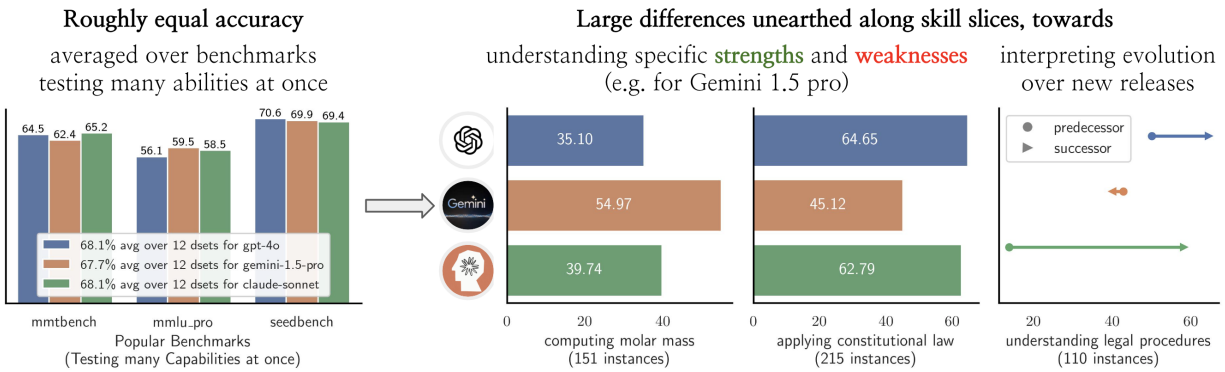


Figure 7.1: We leverage model-generated rationales to extract the skills relevant to any evaluation instance. Inspecting accuracy along *skill-slices* (instances drawn across benchmarks involving the same skill) surfaces fine-grained insights otherwise obfuscated by aggregate accuracy.

by attributes to extract image classification failure modes (e.g. 'fails on *white* foxes') [189, 190], but these methods fail to produce deeper insights on the underlying competencies that models use across contexts (e.g. 'fails on *counting*').

To this end, we propose a method to automatically discover *skills* related to any evaluation instance, toward a finer-grained understanding of model capabilities from existing benchmarks. We consider skills as latent features pertaining to "how" a model must operate – the steps involved in solving a given task – whereas a general attribute relates to "what" is being addressed – the observable characteristics of the instance. Moreover, the relevant skills for an instance may not be readily apparent through superficial examination, often only revealing themselves upon inspection of the solution. This complexity highlights the challenge of automatically inferring relevant skills and underscores the need to develop new approaches for their effective identification.

To this end, we propose to harness the *rationale*-generating ability of strong models to infer skills, as rationales (i.e. explained solutions) elucidate the steps involved in performing a task, and have had many benefits in the past [191, 192, 193, 194, 195, 196]. In our work, we use

rationales not to directly analyze the rationale-generating model, but instead to annotate *data*, on which *any* model can be evaluated.

Specifically, given an evaluation instance, we instruct a strong model (e.g. GPT-4o) to generate a detailed step-by-step rationale, along with the skill applied in each step. We then parse skills per instance for numerous benchmarks and aggregate instances along ***skill-slices***, i.e. subsets of instances drawn from multiple benchmarks that all involve a specific skill, enabling measurement of model proficiency for that skill. After applying our technique to over $45k$ instances from a dozen popular modern benchmarks (primarily multimodal), we find *hundreds* of skills with slice sizes of ≥ 100. To validate the relevance of our skills, we devise an automatic verification protocol and observe that our automatically extracted skills are relevant, even in cases where the annotating model fails in solving the query problem. We will release all our rationales, skill annotations, and skill-slices, which we term the *Skill-Index*, to the public.

Having verified the accuracy of our constructed skill-slices, we present multiple ways our skill annotations and skill-slices can be used to inform our understanding and usage of current large models. First, we conduct detailed skill-level analyses of 6 state-of-art models from the GPT, Gemini, and Claude families, **identifying practical insights into model strengths and weaknesses at previously underexplored granularities**. Despite roughly equal average accuracy across our corpus for GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet, we discover skills for each model where accuracy far exceeds that of its counterparts, with gaps as large as 20% (see Figure 7.1). For example, we find Gemini 1.5 Pro to be much stronger in math and science skills, like 'computing molar mass', while falling far behind for legal skills like 'applying constitutional law'. Next, we show that our **skill-slices allow for fine-grained understanding of improvements in new models within a model family**: while Gemini 1.5 Pro did not improve over v1.0

on 'understanding legal procedures', OpenAI and Claude models improved substantially, with Claude 3.5 Sonnet leaping nearly 50% over Claude 3 Opus.

Further, our **skill-level understanding of models can be directly utilized for instance level model selection** based on skills required. When routing each query instance to a model with best performance on relevant skills, we see improvements in overall accuracy across a dozen benchmarks by 3%, including gains of 3.5 to 7% on MMLU Pro [181]. Finally, to assess a model's proficiency on a skill in isolation, we generate probing questions that directly target a specific skill. We find that model performance on the synthetic probing sets corroborates accuracies on our constructed skill-slices, further strengthening our findings.

Our skill based framework and findings advocate for a paradigm shift in foundation model evaluation, emphasizing the importance of skill-specific analysis to gain a more granular and actionable understanding of model capabilities. We summarize key contributions below:

- Automated Skill Inference: We present a scalable method that leverages model-generated rationales to recover (and validate) the underlying skills relevant to any evaluation instance.
- Novel findings from *Skill-Slices*: Analyzing skill-slices – sets of instances testing a common skill – reveals new fine-grained insights on the trade-offs across leading models.
- Skill Annotations for Popular Benchmarks: We release the *Skill-Index*, a dataset of instance-level skill annotations and rationales, along with code to easily expand to new benchmarks, providing a valuable resource for the research community.

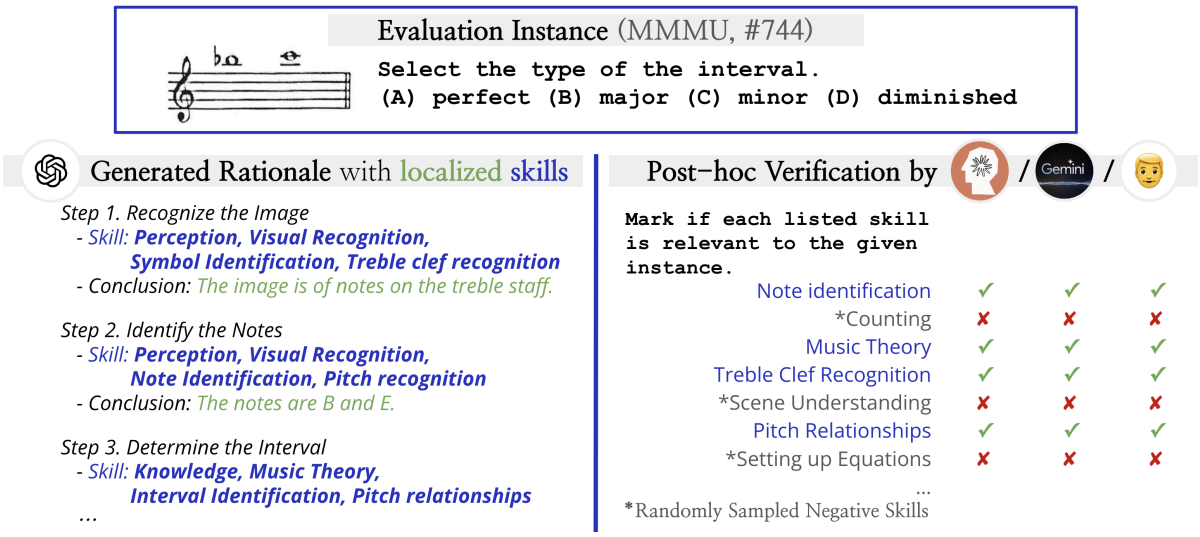


Figure 7.2: **(left)** Sample GPT-4o generated rationale: each skill is listed under multiple names of *cascaded granularity*, and localized to a specific step and concluding claim. **(right)** Annotated skills can be verified independently with a second model or human. We include randomly sampled negative skills and multiple verifiers to assure the quality of the verification, as detailed in 7.2.3.

7.2 Constructing skill-slices using Model-Generated Rationales

We now detail our method for inferring relevant skills for each evaluation instance and aggregating skill annotations across datasets to form *skill-slices* (i.e. a set of instances that share a relevant skill). We utilize *rationales*, or step-by-step solutions, generated by a strong model (GPT-4o) to facilitate the extraction of relevant skills. However, as evaluation instances are challenging by design and generating relevant skills requires a significant level of meta-reasoning, it is important to assess the quality of produced skills. Thus, we additionally present a multifaceted automatic approach to validating the relevance of extracted skills, which we confirm aligns well with human judgements.

7.2.1 Skill Extraction via Rationale Parsing

Given an evaluation instance, we prompt GPT-4o to generate a detailed rationale (i.e. step-by-step solution), where each step only involves the use of a single skill; figure 7.2 presents an example. We additionally instruct the model to, after each step, list the applied skill with *multiple* names of *cascading granularity*. For example, the skill 'treble clef recognition' is prepended with 'perception, visual recognition, symbol identification'[1]. Listing multiple names of different granularity per skill is a simple way to increase the number of slices an evaluation instance falls into, leading to a greater quantity and diversity of skill-slices. Further, each granularity has unique and complementary advantages: fine-grained skills are more specific, while coarse-grained skills lead to slices with a larger number of samples, making an accuracy estimate over such slices more reliable. Also, in order to standardize the response structure, we include an in-context example in our prompt, which enables simple parsing of rationales to extract skills.

Note that other methods could possibly be used to annotate relevant skills, like directly prompting GPT-4o to list them. We prioritized maximizing the total count and size of resultant skill-slices. In comparing the skills per instance obtained by direct prompting vs. our rationale parsing, we indeed find that rationale parsing results in a higher average count and diversity in grain.

[1] If some of these skills are unfamiliar to you, you're not alone! Many benchmarks today ask questions that are challenging to anyone without expertise in the relevant domain. This precisely motivates relying on stronger models to help us make sense of these benchmarks, as manual inspection has increasingly limited utility.

7.2.2 Curating Skill-Slices across Benchmarks

With rationale-parsing, we tag relevant skills for 46k evaluation instances from 12 benchmarks that are popular (i.e. commonly featured in evaluations for recent model releases) and unsaturated (SOTA under 90% accuracy), resulting in 690k total skills (128k unique) over 202k rationale steps. Importantly, we observe that **skills cut across benchmarks**. That is, many skill-slices of non-trivial size are formed when including instances across benchmarks, because *models re-use skills in diverse contexts*. Namely, 278 skill-slices have at least 100 unique instances each. This number rises to 332 after de-duplicating via a tight clustering[2] on the text embeddings of the skills. Aggregating across benchmarks enables us to analyze skills that are in the long tail for one benchmark, but get resurfaced when many different benchmarks are joined together. Indeed, slice count increases by 51% when slicing across benchmarks instead of only drawing instances from one benchmark at a time.

7.2.3 Automated Validation of Skill Relevance

In the absence of fine-grained ground truth skill annotations that would confirm whether generated skills are relevant or not, we design an automatic validation approach. Namely, we propose two automated methods to directly validate the relevancy of listed skills: *post-hoc verification* and *inter-(skill)annotator agreement*. The latter checks the overlap in skills listed for the same instance by two different annotators. Here, we focus on the former: post-hoc verification consists of a second 'verifier' model which inspects an evaluation instance with a list of skills and marks whether each skill is relevant or not. We form the list of skills by combining the skills

[2]We employ a high minimum cosine similarity threshold of 0.95 per cluster to de-duplicate skills listed under slightly different names; e.g., 'trigonometry' and 'trigonometric calculation' have a similarity of 0.95.

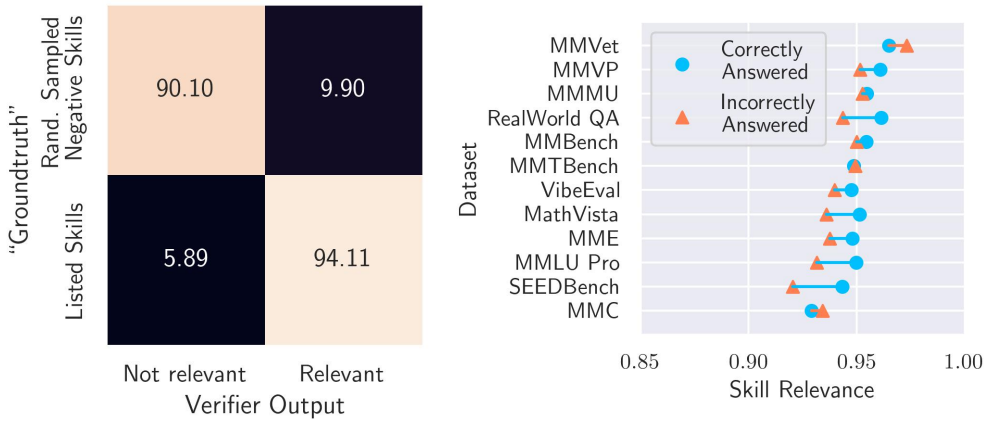


Figure 7.3: (**left**) Post-hoc verification shows GPT-4o-annotated skills are relevant, and that automatic verifiers are reliable, as they admit low rates of false positives (marking a randomly sampled negative skill as relevant). (**right**) Rate that GPT-4o-annotated skills are marked as relevant, separated by if GPT-4o correctly answered the underlying evaluation instance (blue) or not (orange). Empirically, annotated skills have high relevancy rates *even* when the annotator incorrectly answers the question.

for the instance with an equally sized list of *negative* skills, randomly sampled as follows: with S denoting all skills from our corpus, negative skills for an instance x with skills S_x are drawn from $\{s \in S \mid \max_{s_i \in S_x} \text{sim}(s, s_i) \leq \tau\}$. Here, 'sim' is the similarity of two skills, computed via the cosine similarity of text embeddings from a frozen text embedder. The threshold τ is selected based on the text embedder so that no negative skill is equivalent up to paraphrasing to an annotated skill for the instance. We include negative skills as quality controls to ensure that the verifier model does not simply mark all skills as relevant.

Figure 7.3 shows the results of post-hoc verification of skills for 100 skills per dataset averaged over three verifier models (Claude 3.5 Sonnet, Gemini 1.5 Pro, GPT-4v[3]). We observe 94.1% of skills annotated by GPT-4o are verified as relevant. A smaller human validation of 640 total skills results in 95.7% relevancy rate, as well as 92.8% agreement with automatic verifiers, indicating that post-hoc verification is a reliable automated metric for assessing skill relevancy.

[3]GPT-4v denotes GPT-4 Turbo 2024-04-09, and GPT-4o denotes GPT-4o 2024-05-13 throughout our paper.

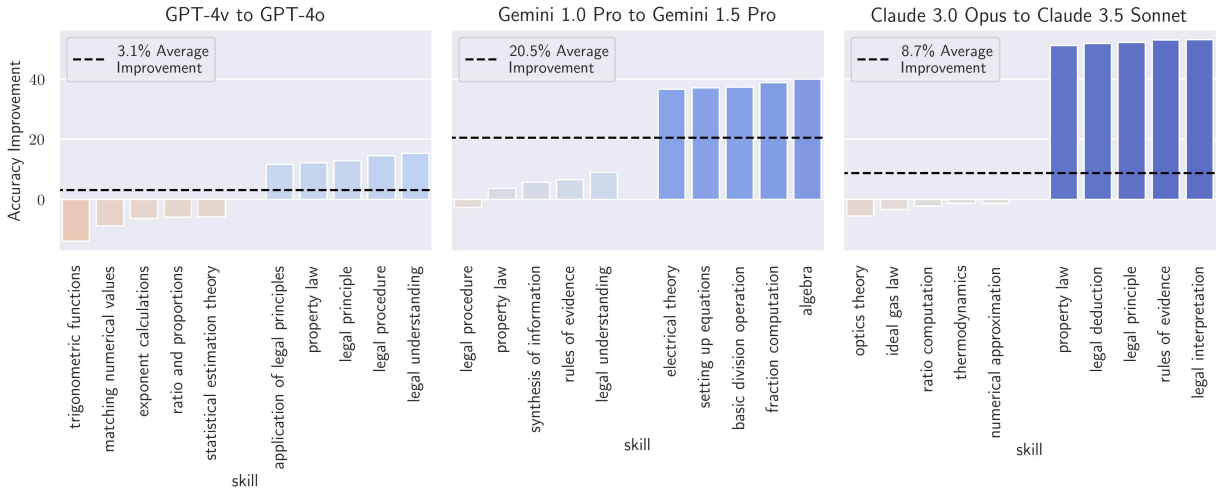


Figure 7.4: Skill-slices shed insight on how models evolve over new releases. For GPT and Claude models, skills related to law see the largest increases in accuracy, while Gemini models improve most in performing math and science skills.

The right panel shows that the relevancy rate is roughly equivalent, *regardless of if the annotator correctly answers the underlying evaluation instance*; crucially, this enables using automated skill annotations on instances that are interesting to evaluate on (i.e. hard enough to stump some models). Thus, while rationales may at times be partially incorrect[4], they can still be leveraged to shed insight on *data* at scale.

7.3 Analyzing Foundation Models with Skill-slices

We now leverage the skill-slices obtained in the previous section to better understand and utilize frontier models from OpenAI, Google, and Anthropic. First, we identify skill-slices with vastly different accuracies across releases from the same family, as well as across families, which are obfuscated when inspecting overall accuracy. Then, having discovered models differ in their specific strengths and weaknesses, we show how overall accuracy can be improved by choosing the best model per instance or dataset, leveraging inferred skills and computed model-wise skill

[4]In fact, there's an error in the rationale in Figure 7.2: the notes are $B\flat$ & C. However, the skills are accurate.

accuracies.

7.3.1 Finer-grained Insights from Existing Evaluation Instances

Understanding Model Evolution. Figure 7.4 shows average improvement, along with skill-slices with greatest and least improvement, for the last two releases in the Open AI GPT, Google Gemini, and Anthropic Claude families[5]. For all model families, we find skills where improvement is more than twice the average, as well as skills where the model does not improve at all. Notably, both GPT and Claude models see greatest increases in skills related to law, with the Claude-Sonnnet 3.5 improving over Claude-Opus 3.0 by a staggering $\sim 50\%$ along numerous law skill-slices, suggesting improving legal abilities of their models may have been a recent priority for OpenAI and Anthropic.

Finding 1: **Skill-level improvements across model releases greatly differ (i) from average improvement and (ii) between families.** The biggest leaps for GPT and Claude families were for legal skills, while Gemini improved most for math and science skills (figure 7.4).

Interpreting Trade-offs. We now directly compare GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet on all skill-slices with at least 100 unique evaluation instances. *While overall accuracies for the three models across our evaluation corpus falls within a half percent of one another, we observe differences as high as* 25% *for certain skills*, as shown in figure 7.5. Further, some patterns emerge: Gemini 1.5 Pro is distinctly strong for math and science, while GPT-4o's rela-

[5]Because Gemini-1.0-pro is not multimodal, we present results for skill-slices curated from $12k$ language-only evaluation instances (instead of our entire corpus) with at least 100 instances within each slice.



Figure 7.5: Unique strengths (**top**) and weaknesses (**bottom**) of GPT-4o, Gemini 1.5 Pro, and Claude 3.5 Sonnet, relative to one another. We present skills where each model's slice accuracy is highest / lowest (respectively) relative to the average of the other two model accuracies. † denotes significance ($p < 0.01$) via Wilcoxon signed-rank test; †† means both gaps vs. the other models are significant.

tive strengths pertain to visual skills like color differentiation and object localization, as well as the real-world skill of *traffic signal identification*, where GPT-4o is on average 16.7% more accurate. Interestingly, Gemini 1.5 Pro's greatest relative weaknesses nearly all concern law, which we found above to be the key skills that GPT-4o and Claude 3.5 Sonnet improved upon compared to their respective predecessors.

Finding 2: **Models have unique strengths and weaknesses**: e.g., GPT-4o excels in visual skills, but lags behind on various math skills. Conversely, Gemini 1.5 Pro outperforms both other models on math and science skills, while falling far behind for legal skills (figure 7.5).

Unfortunately, since we have limited visibility into the training data and processes for these private models, we cannot offer explanations for the strengths and weaknesses we observe. Nonetheless, this illuminates an advantage of our method: by annotating evaluation instances, **skill-slice analysis enables fine-grain insight on model capabilities, even when only granted black-box access**.

7.3.2 Generalization of Skill-Slice Insights

Our skill-slice analysis operates under the premise that the insights drawn from inspecting sufficiently-large slices from a wide range of sources can *generalize* to new instances. If the insights generalize well, then understanding skill trade-offs between models can enable employing the models in a more calculated manner. That is, if we have knowledge of the skills relevant to a new instance, as well as each model's skill-wise strengths, we can improve accuracy by *routing* that instance to the model whose strengths are most aligned with those skills.

To test this, we route each instance in our corpus to one of GPT-4o, Gemini 1.5 Pro, and

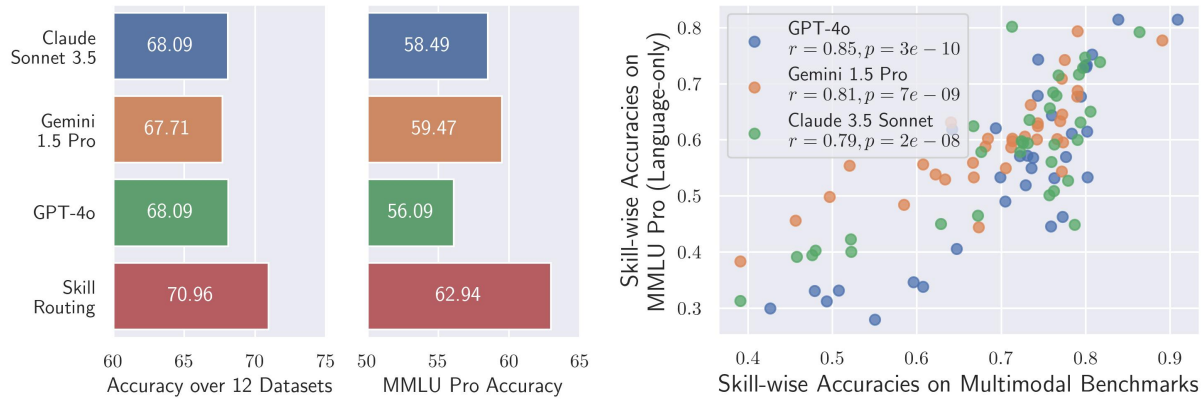


Figure 7.6: (**left**) A simple routing scheme that assigns each instance to the model with best accuracies for the skills relevant to that instance can lead to noticeable accuracy gains. (**right**) Accuracies over skill-slices can generalize to unseen data. In fact, skill-wise accuracies computed over multimodal benchmarks correlate strongly with accuracies obtained over a language-only benchmark.

Claude 3.5 Sonnet, based on the skill annotations for that instance and the skill-wise accuracies per model computed over the remaining corpus (i.e. without the test instance). To obtain a single score per instance per model, we take a weighted average of skill-wise accuracies, where the weight for each skill is the inverse of its slice size (so to upweight finer-grained, more specific skills). As shown in the left panel of figure 7.6, routing increases accuracy by up to 3.2% compared to each of the frontier models alone on 12 datasets combined, including improvements of 3.5 to 6.8% for MMLU Pro [181].

We highlight the MMLU Pro results because the vast majority of the instances in the reference corpus (over which skill-wise accuracies are computed) are from multimodal benchmarks, while MMLU Pro is language-only, making it a good candidate to showcase the generalization of our approach. To study this deeper, we now partition each skill-slice based on if the instance comes from MMLU Pro or one of our multimodal benchmarks. Then, for each model we obtain two paired sets of slice-wise accuracies: one accuracy score per skill-slice, per partition. As shown in the right panel of figure 7.6, we observe strong correlations ($r \geq 0.79$, $p < 1e-7$)

between these two sets of scores for all three models. In addition to showing generalization, this result further validates our idea that skills pertain to a deeper property of a given instance (namely, what must be done to *solve* it) than surface level attributes, like its modality.

Finding 3: **Skill-slice accuracies enable instance-wise model selection**. Routing each instance to the model strongest on the relevant skills results in a 3% accuracy gain, indicating that skill-slice analyses offer generalizable insight (see figure 7.6).

We note that this proof of concept utilizes accuracies computed over slices defined by a single skill, not accounting for the added difficulty created when certain skills are used together. Nonetheless, our framework opens the door to studying these second order effects, e.g., by surfacing error-inducing slices defined by a *combination* of skills, as done by [190] for image classifiers.

7.4 Corroborating Slice Accuracies with Probing Questions

While skill-slices allow for approximating a model's proficiency at a skill by averaging accuracy over any instance where the skill is relevant, our rationale parsing framework enables a second independent analysis to more directly probe a *single* skill, without the effect of co-occurring skills.

Namely, rationale parsing *localizes* each skill to a specific step of the generated solution, as well as the resulting claim, which we instruct the rationale-generating model to include in its response. For example, as shown on the right in figure 7.7, the resultant claim to a step where the skill "[musical] note identification" is applied is "The notes are B and E". A claim can then

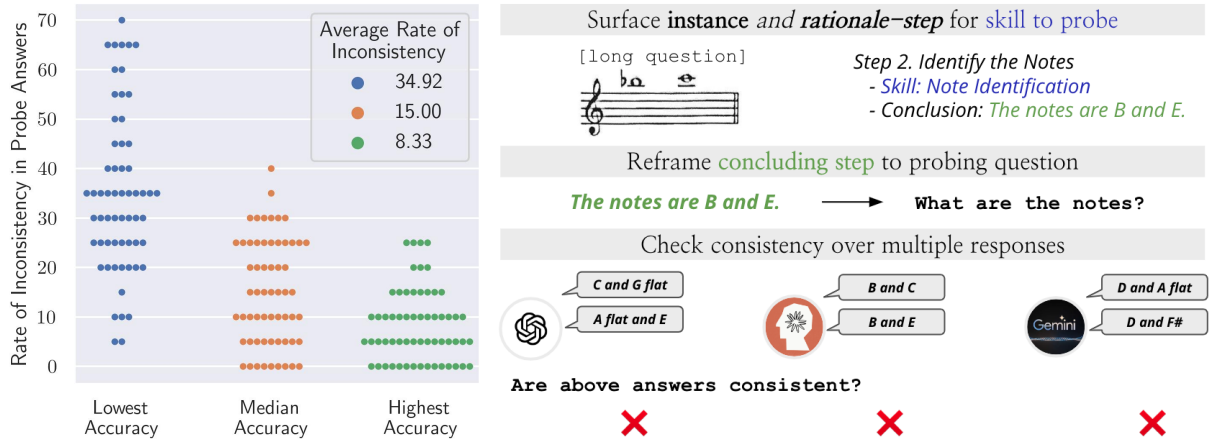


Figure 7.7: (**left**) Skills who's slices yield lowest model accuracy have the highest rates of inconsistency when probed. Each point corresponds to one model's consistency rate for a single skill. (**right**) Probe questions are generated by reframing rationale steps associated with a skill of interest.

be reframed as a question (e.g. "What are the notes?") probing a single skill, unlike the original question that often requires numerous skills and steps, any one of which could cause an incorrect answer. We use the rate of *inconsistency*, like in [197], over multiple responses to each probing question as a measure of proficiency at the probed skill: if a model contradicts itself, it must have been wrong at least once.

We now probe three sets of 20 skills: those with lowest, median, and highest accuracy based on the analysis in section 7.3, to sample skills with a variety of slice accuracies. For each skill, we (i) generate 20 probing questions, (ii) obtain 5 responses per probe per model, and (iii) evaluate consistency of the responses with GPT-4o[6]. As shown in figure 7.7, models contradict themselves at a significantly higher rate for low accuracy skills, and **inconsistency rises as slice accuracy falls**, corroborating the skill-slice analysis. Overall, inconsistency correlates well with slice accuracy ($r = -0.675$). Further, we find skills where all models contradict themselves more

[6]While we use GPT-4o, forming probing questions and evaluating consistency are likely simple enough to be done reliably by open-source LLMs (as both tasks are text-only).

often than not, such as "[musical] note identification", "eye direction analysis", and "enumerating objects", the last of which is a well-documented limitation of VLMs [198, 199].

Finding 4: **Rationale-parsing enables targeting skills in isolation.** This independent analysis yields corroborating assessments of skill proficiency as skill-slice accuracies (figure 7.7).

In addition to independently verifying the insights from slice accuracies, probe inconsistency offers *complementary* signal: since some evaluation instances engage multiple skills, low accuracy over one skill-slice could be due to a deficiency on another highly co-occurring skill. However, if a model is proficient in a skill, it will always answer consistently (correctly) to probing questions. Thus, consistency on probing questions can be used to *refine* insights from skill-slice analyses, as a skill with low slice accuracy and high probe inconsistency is almost surely deficient.

7.5 Retrieving Evaluation instances for Custom Query Skills

Finally, towards (a) direct comparison of our work to prior methods and (b) more flexible use of skill annotations, we introduce the task of *skill-based retrieval*: given an open-vocabulary query skill, can we retrieve relevant evaluation instances, so to build a custom skill-slice for specialized evaluation? This task boils down to defining an efficient metric that assigns a similarity score between a text query and an evaluation instance. Here, we focus on multimodal evaluation instances[7].

We consider **baselines** that leverage **a.** embeddings of the instance (either of just the image

[7] A multimodal instance consists of one **image** along with a **text** question.

using CLIP [13], just the text question, or the average of both image and text), **b.** attributes of the input, either inferred by GPT-4o[8] or provided as ground-truths by benchmark creators. In this latter case, we embed text attributes, along with the query skill, using a text encoder, with which we compute similarity of an attribute to the query skill as the cosine similarity of their embeddings. To obtain a single similarity score for the GPT-4o attributes baseline where multiple attributes exist per instance, we average the top 3 highest similarities[9]. Our method is identical to the attribute-based approach, except that we leverage annotated *skills* instead of attributes. The baselines encapsulate how prior methods propose to group inputs to go beyond overall accuracy [189, 190], enabling direct comparison to our method.

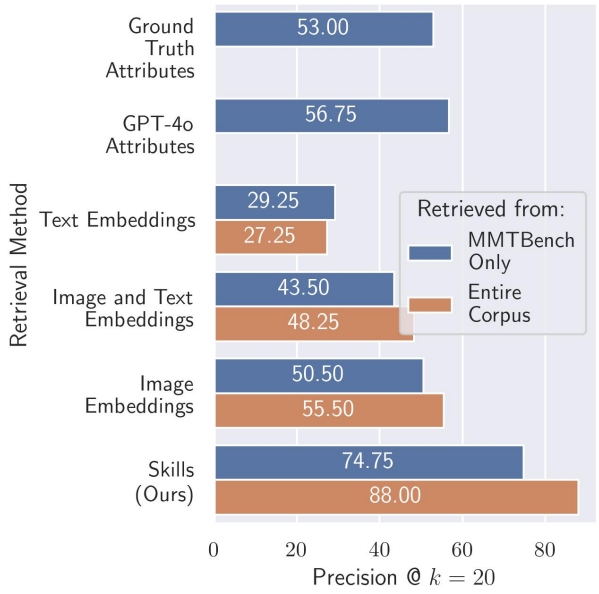


Figure 7.8: Retrieving instances based on a query skill is far more effective when utilizing skill annotations than using direct representations of each instance, as done in prior work.

Setup: We take 20 skills from one benchmark (MMBench), and retrieve 20 instances per skill per method from two sources: **a.** a single other benchmark: MMTBench, selected because it has the highest number of ground-truth categories (162) out of our corpus, **b.** all benchmarks aside from MMBench, though we exclude ground-truth and GPT-4o inferred attributes baselines in the second case, for availability and cost reasons respectively.

Metric: We evaluate each retrieved instance by presenting it along with the query skill to GPT-

[8] We obtain surface-level 'tags': the type of image, subject of the question, etc.
[9] As in [200], we find top-3 averaging to be superior to top-1 or full averaging.

4o and prompting it to decide if the query skill is relevant or not; 'precision @ $k = 20$' is the fraction of the 20 retrieved instances deemed relevant.

As shown in figure 7.8, retrieval using skill annotations far exceeds the precision for any other baseline in both settings, with improvements of 18-45.5% and 50.75-32.5% respectively. The attribute baselines are stronger than the embedding-based approaches, though obtaining attributes is also more expensive. The embeddding-based baselines struggle, reflecting that frozen-embedding models may prioritize encoding surface-level information over skills, as skills may only become more apparent when inspecting the *solution* to the instance. We also find that searching over a broader corpus benefits the skill-based approach far more than either embedding-based approach. In summary, while allowing for direct searching over our corpus, these skill-based retrieval experiments underscore how our skill annotations offer novel signal, enabling the grouping of instances in ways that can be complementary to prior methods.

Finding 5: **Skill annotations open the door to custom evaluation sets via skill-based retrieval.** Similarity of attributes or direct embeddings for two instances does not reflect the similarity of the underlying skills tested by each instance, leading to poor skill-based retrieval performance (figure 7.8).

7.6 Review of Literature

Prior efforts to gain **fine-grain insights** often involve intensive manual efforts to construct multi-task benchmarks [201, 202, 203, 204, 205] or datasets with annotations beyond ground-truth labels [206, 207, 208, 209]. On top of quantifying specific abilities, extra annotations have illuminated biases [9, 210] and robustness issues [73, 211]. Automated alternatives include syn-

thesizing benchmarks with known attributes [212, 213] or automatically uncovering sub-groups within existing data [214, 215].The term *slice discovery* (or otherwise referred to as data *cohort* or *subgroup*) was coined [189, 216] to describe these methods which enable comparisons [217] and error analysis [48, 190, 218, 219, 220]. While these works focus on slices formed by surface-level attributes for visual recognition tasks, we curate *skill-slices*, which pertain to the underlying ability each slice requires from a model, and as such extend to more tasks and modalities.

Skills are akin to the 'ability' tags present in some modern benchmarks [185, 186], though they can be finer (or coarser) grained [221] than what is typically annotated. The importance of acquisition of skills has been studied in human cognition [222, 223] as well as language models [224]. [225] utilizes skills as input to an LLM to generate challenging evaluation instances requiring a composition of the provided skills. Recent [215] and concurrent [226] work also explore skill inference, albeit in more narrow domains, toward improved prompting or finetuning. This important shift of methods from attributes to skills is indeed motivated by the increasingly general-purpose nature of state-of-the art models. In this work, we scale up such analysis and insights by expanding it to 12 benchmarks and by introducing *rationale parsing* as a means to infer more skills.

Rationales have been studied extensively in the context of prompting [191, 192, 227]. Other works show the value of rationales as richer training signal [195, 196, 228, 229] or model explanations [194, 230, 231], though some question their faithfulness to underlying model processes [232, 233]. We utilize rationales not to interpret a model directly, but instead to improve the relevance and diversity of the annotated skills [193], which become apparent in the solution steps present in rationales.

7.7 Discussion and Future Work

It is often said, "We can only improve what we can measure." As models grow in complexity and capability, traditional evaluation metrics like aggregate accuracy over entire datasets become insufficient. Simply averaging accuracy across datasets hides important insights needed to understand and enhance both the datasets and the models themselves.

To address this gap, our work introduced a skill-slice analysis approach for model evaluation. By examining model-generated rationales, we developed an automated method to extract the underlying skills required to solve individual evaluation instances. This shift from focusing on overall accuracy to conducting a more detailed analysis provides a more granular picture of a model's strengths and weaknesses. Furthermore, our reliance on a strong model to help humans understand evaluation data may prove to be increasingly pertinent as models get so strong that only a small handful of people will be qualified to provide manual annotations on evaluation instances. In this sense, our work relates to scalable oversight [234], providing a new avenue to evaluation leveraging human-AI collaboration to understand models that will encapsulate advanced skills and knowledge.

Looking ahead, our approach can assist in creating more focused, skill-based evaluation datasets, by accelerating the discovery of skills where models are collectively deficient or exhibit disparate performance. Our skill-based retrieval method can play a pivotal role in generating valuable training data for model development. By concentrating on specific skills, one can create more effective training sets that address identified weaknesses, thereby accelerating the advancement of AI models. This aligns with the principle that meaningful progress is achievable when we have clear insights, reinforcing the notion that we can only improve what we can measure.

Chapter 8: Prioritizing Interpretability in Failure Mode Extraction

In the previous chapter, we leveraged an auxiliary model to obtain a new kind of annotation: skills, as opposed to surface-level attributes. Equipped with these annotations, we could inspect model performance on curated subsets of data to obtain skill-wise accuracies. However, skill-wise accuracy is limited in measuring the challenge posed when a model is required to apply two or more skills at once. In this chapter, we return to the simpler setting of image classification, where we demonstrate that even richer insights can be gained when curating data subsets defined by the presence of a combination of attributes. Namely, we present an automatic method to efficiently form and search over the space of use cases characterized by a combination of attributes. At the heart of our method is the reliance on an auxiliary model to obtain interpretable attributes, which leads to more faithful and coherent descriptions of discovered failure modes compared to existing methods. This ICLR '24 work was led by Keivan Rezaei and Mehrdad Saberi, and co-authored by myself and Soheil Feizi [235].

8.1 Summary

A plethora of reasons (spurious correlations, imbalanced data, corrupted inputs, etc.) may lead a model to underperform on a specific subpopulation; we term this a *failure mode*. Failure modes are challenging to identify due to the black-box nature of deep models, and further, they

are often obfuscated by common metrics like overall accuracy, leading to a false sense of security. However, these failures can have significant real-world consequences, such as perpetuating algorithmic bias [9] or unexpected catastrophic failure under distribution shift. Thus, the discovery and description of failure modes is crucial in building reliable AI, as we cannot fix a problem without first diagnosing it.

Detection of failure modes or biases within trained models has been studied in the literature. Prior work [236, 237] requires humans in the loop to get a sense of biases or subpopulations on which a model underperforms. Some other methods [238, 239, 240, 241] do the process of capturing and intervening in hard inputs without providing *human-understandable* descriptions for challenging subpopulations. Providing human-understandable and *interpretable* descriptions for failure modes not only enables humans to easily understand hard subpopulations, but enables the use of text-to-image methods [242, 243, 244, 245] to generate relevant images corresponding to failure modes to improve model's accuracy over them.

Recent work [168, 189, 246, 247] takes an important step in improving failure mode diagnosis by additionally finding natural language descriptions of detected failure modes, namely via leveraging modern vision-language models. These methodologies leverage the shared vision-language latent space, discerning intricate clusters or directions within this space, and subsequently attributing human-comprehensible descriptions to them. However, questions have been raised regarding *the quality of the generated descriptions*, i.e., there is a need to ascertain whether the captions produced genuinely correspond to the images within the identified subpopulation. Additionally, it is essential to determine whether these images convey shared semantic attributes that can be effectively articulated through textual descriptions.

In this work, we investigate whether or not the latent representation space is a good proxy

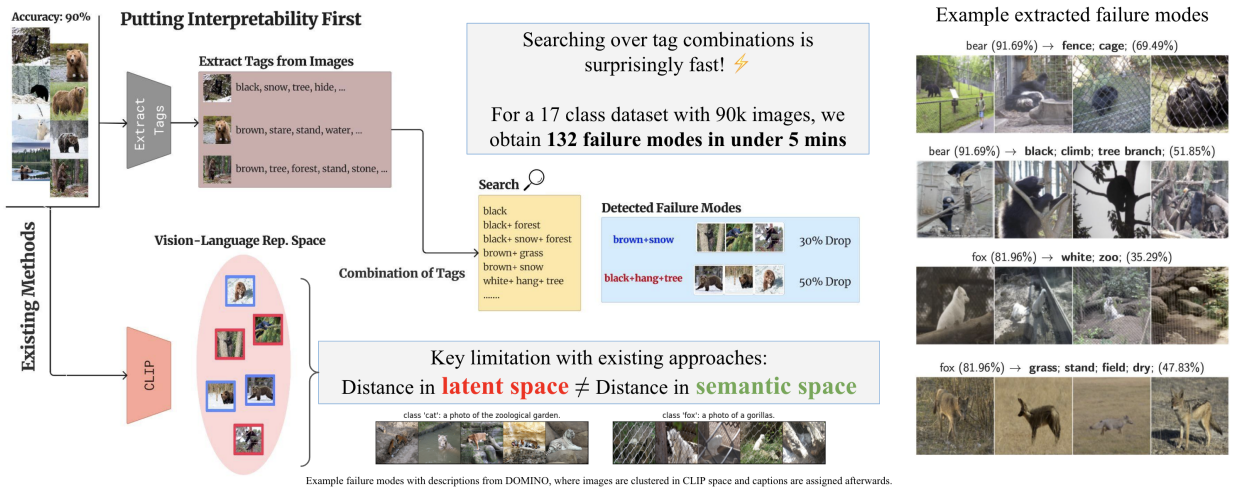


Figure 8.1: Overview of PRIME. Prior methods cluster inputs in an uninterpretable activation space, leading to semantically misaligned failure mode descriptions. PRIME utilizes an auxiliary model to obtain faithful attributes, before efficiently combining them to extract specific and accurate descriptions.

for semantic space. In fact, we consider two attributed datasets: CelebA[33] and CUB-200[248] and observe that two samples sharing many semantic attributes may indeed lie far away in latent space, while nearby instances may not share any semantics. Hence, existing methods may suffer from relying on representation space as clusters and directions found in this space may contain images with different semantic attributes leading to less coherent descriptions.

Inspired by this observation and the significance of faithful descriptions, we propose PRIME. In this method, we suggest to reverse the prevailing paradigm in failure mode diagnosis. That is, we put *interpretability first*. In our method, we start by obtaining human-understandable concepts (tags) of images using a pre-trained tagging model and examine model's behavior conditioning on the presence or absence of a combination of those tags. In particular, we consider different groups of tags and check whether (1) there is a significant drop in model's accuracy over images that represent all tags in the group and (2) that group is minimal, i.e., images having only some of those tags are easier images for the model. When a group of tags satisfies both of these condi-

tions, we identify it as a failure mode which can be effectively described by these tags. Figure 8.1 shows the overview of our approach and compares it with existing methods.

As an example, by running PRIME on a trained model over Living17, we realize that images where a **black** ape is **hanging** from a **tree branch** identify a hard subpopulation such that model's accuracy drops from 86.23% to 41.88%. Crucially, presence of all 3 of these tags is necessary, i.e., when we consider images that have 1 or 2 of these 3 tags, the accuracy of model is higher. Figure 8.2 illustrates these failure modes.

To further validate our method, we examine data unseen during the computation of our failure mode descriptions. We observe that the images that match a failure mode lead the model to similarly struggle. That is, we demonstrate *generalizability* of our failure modes, crucially, directly from the succinct text descriptions. While reflecting the quality of our descriptions, this allows for bringing in generative models. We validate this claim by generating hard images using some of the failure mode's descriptions and compare the accuracy of model on them with some other generated images that correspond to easier subpopulations.

Finally, we show that PRIME produces better descriptions for detected failure modes in terms of *similarity*, *coherency*, and *specificity* of descriptions, compared to prior work that does not prioritize interpretability. Evaluating description quality is challenging and typically requires human assessment, which can be impractical for extensive studies. To mitigate that, inspired by CLIPScore [249], we present a suite of three automated metrics that harness vision-language models to evaluate the quality. These metrics quantify both the intra-group image-description similarity and coherency, while also assessing the specificity of descriptions to ensure they are confined to the designated image groups. We mainly observe that due to putting interpretability first and considering different combinations of tags (concepts), we observe improvements in the

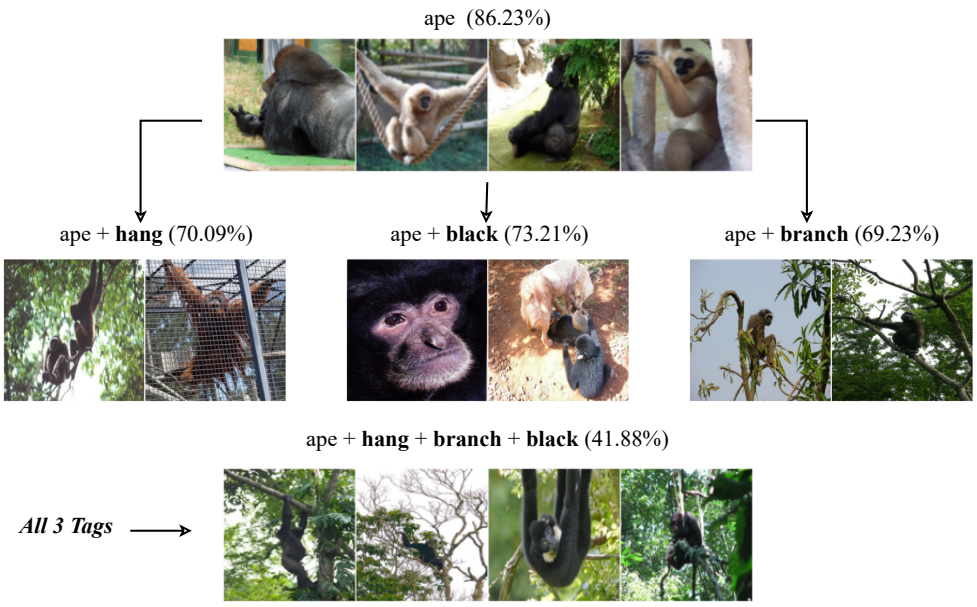


Figure 8.2: Although appearance of tags "hang", "black", and "branch" individually lowers model's accuracy, when all of them appear in the images, model's accuracy drops from 86.23% to 41.88%.

quality of generated descriptions.

Summary of Contribution.

1. We propose PRIME to extract and explain failure modes of a model in human-understandable terms by prioritizing interpretability.

2. Using a suite of three automated metrics to evaluate the quality of generated descriptions, we observe improvements in our method compared to strong baselines such as [189] and [168] on various datasets.

3. We advocate for the concept of putting interpretability first by providing empirical evidence derived from latent space analysis, suggesting that distance in latent space may at times be a misleading measure of semantic similarity for explaining model failure modes.

Chapter 9: Spuriosity Rankings: Sorting Data to Measure and Mitigate Biases

Having closely studied spurious correlations in vision models and proposed a method to scalably find and describe the failure modes they cause, we now take another step in this direction with *Spuriosity Rankings*. This chapter, detailing a NeurIPS '23 spotlight paper co-authored with Wenxiao Wang, Sahil Singla, and Soheil Feizi [250], again makes use of an auxiliary model to better organize data. Namely, we utilize an adversarially trained model to uncover relevant spurious cues per class, as well as – importantly – subsets of data where the cues are strongly present or absent. This work then, in addition to discovering failure modes, actually provides mitigation as well, like the first chapter of this section. Without further ado, we present Spuriosity Rankings, the culmination of our foray into spurious correlations in image classifiers.

We present a simple but effective method to measure and mitigate model biases caused by reliance on spurious cues. Instead of requiring costly changes to one's data or model training, our method better utilizes the data one already has by sorting them. Specifically, we rank images within their classes based on spuriosity (the degree to which common spurious cues are present), proxied via deep neural features of an interpretable network. With spuriosity rankings, it is easy to identify minority subpopulations (i.e. low spuriosity images) and assess model bias as the gap in accuracy between high and low spuriosity images. One can even efficiently remove a model's bias at little cost to accuracy by finetuning its classification head on low spuriosity

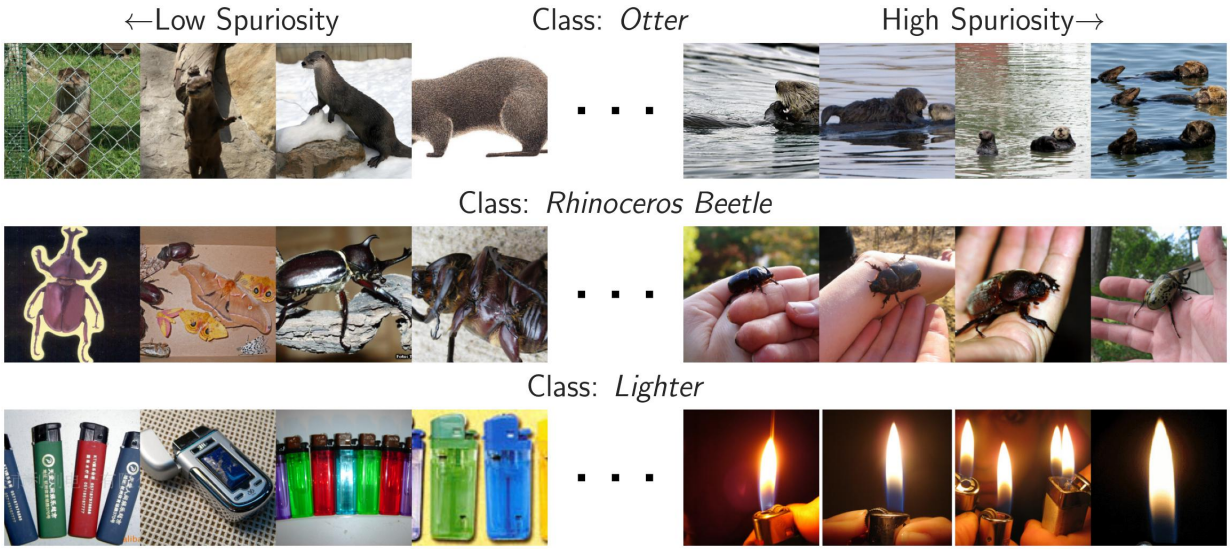


Figure 9.1: Spuriosity rankings sort images based on the presence of spurious cues. The rankings reveal hidden spurious features that models rely on, like *rippling water, human hands,* and *flame in the dark* (right; top to bottom), and uncover minority subpopulations where spurious cues are absent (left).

images, resulting in fairer treatment of samples regardless of spuriosity. We demonstrate our method on ImageNet, annotating 5000 class-feature dependencies (630 of which we find to be spurious) and generating a dataset of $325k$ soft segmentations for these features along the way. Having computed spuriosity rankings via the identified spurious neural features, we assess biases for 89 diverse models and find that class-wise biases are highly correlated across models. Our results suggest that model bias due to spurious feature reliance is influenced far more by what the model is trained on than how it is trained.

9.1 Introduction

Deep models often exhibit bias, underperforming on certain minority subpopulations [9, 120]. One cause of bias is the tendency of deep models to rely on spurious features [6, 7, 117], as samples that retain spurious correlations are predicted more accurately than rarer samples where

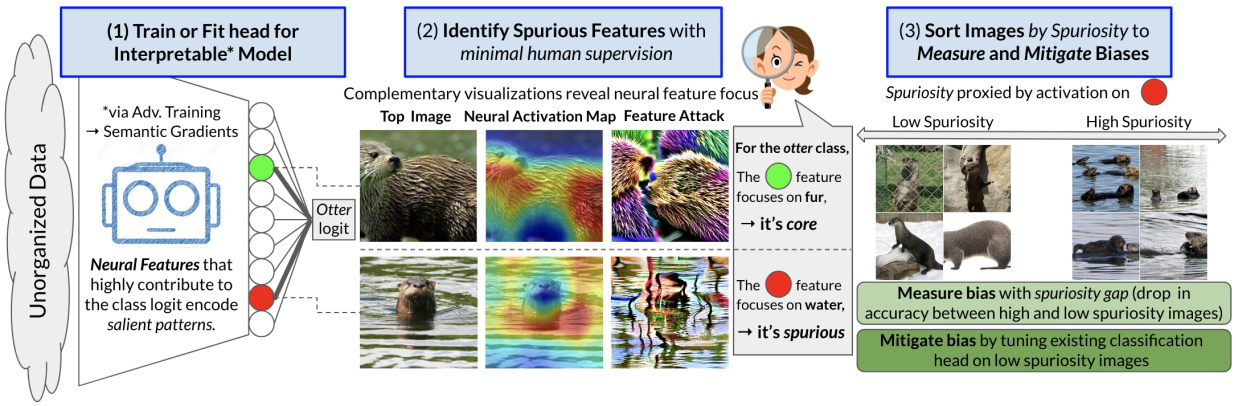


Figure 9.2: Overview of our framework. Per class, we sort images by *spuriosity*, which we proxy via the activation of neural features (i.e. nodes in the penultimate layer of a trained classifier) that focus on spurious cues. After sorting, measuring model bias to the discovered spurious features is as simple as computing the drop in accuracy across high and low spuriosity images. This bias can be mitigated efficiently by tuning the classification head on low spuriosity images.

spurious correlations are broken. For example, a classifier may learn to associate water with the *otter* class if many training images contain water, which in turn can lead to reduced accuracy on images of otters out of water. Spurious feature reliance also hinders robust generalization (e.g. medical systems trained in one hospital may fail when deployed to others [69, 70]), potentially posing major reliability risks.

Recent trends suggest that scaling up models and datasets may resolve many existing limitations [251]. However, in addition to being costly, collecting more data would not alter the long-tail data distribution that leads to spurious feature reliance, as the spurious cues will remain just as correlated, if not more, with class labels. Instead of seeking more data, we suggest to *better utilize the data one already has*. Specifically, we aim to rank images within their classes by *spuriosity*, which can be understood informally as the degree to which relevant spurious cues are present. Figure 9.1 shows examples of our rankings. For the *otter* class, high spuriosity images prominently contain the common spurious feature of water, while low spuriosity images do not, instead displaying the rarer case of otters on land.

Spuriosity, while intuitive, is rather challenging to measure scalably: for each class, one must (i) discover relevant cues (i.e. those that a model will rely upon), (ii) identify the spurious ones, and (iii) obtain a function that scores the level to which these spurious cues are present in a given image. To this end, we propose to extract spurious concept detectors from within an *interpretable model* trained on the given data. Namely, we automatically select neural features (i.e. nodes in the penultimate layer) that contribute highly to a class logit, as they encode concepts that are both salient to that class and relied upon by the model. Then, we interpret the concept each neural feature detects, with limited human supervision, so to keep the ones activated by spurious cues. Finally, we average the activation on these neural features to yield a scalar for each image that proxies the presence of relevant spurious cues (i.e. spuriosity), thus enabling sorting. Once data is ranked by spuriosity, multiple significant benefits emerge immediately: First, low spuriosity images **reveal minority subpopulations**, where common spurious correlations are broken. Second, we can easily **quantify model bias** as the drop in accuracy between high and low spuriosity images; we call this *spurious gap*. Third, we can **mitigate bias** towards high spuriosity images by simply finetuning the classification head on low spuriosity images.

Figure 9.2 diagrams our approach. The interpretable model we use to detect relevant neural features and extract spurious ones (steps 1 and 2 respectively) is adversarially trained, as in [1]. However, we show for the first time that we can extend this framework *without adversarial training*. Namely, to discover patterns salient for a new task, it suffices to merely fit a linear classification head on the new data over fixed adversarially pre-trained neural features. This dramatically increases the efficiency of the method, and expands its use cases to low-data regimes where adversarial training is less feasible. We carry out steps 1 and 2 at an unprecedented scale, resulting in 5000 annotated class-neural feature dependencies over all of the ubiquitous ImageNet, includ-

ing 630 spurious features discovered over 357 classes. For these classes, we compute spuriosity rankings (step 3), which, along with a web-UI[1] for easy viewing and $325k$ soft segmentations of salient visual features in ImageNet, we present as a dataset to shed insight on *how* deep models perform ImageNet classification.

Using spuriosity rankings, we compute *spurious gaps* (drop in accuracy from high to low spuriosity images) for **89 models**, observing that all models underperform on low spuriosity images. Models exhibiting the most and least bias to spurious features are CLIP and adversarially trained models respectively, corroborating prior work [13, 91] and thus further validating our rankings. We also observe diverse models to have highly correlated class-wise spurious gaps. This implies that models absorb the same biases, suggesting that *bias is influenced far less by how a model is trained, and far more by what it is trained on*. Thus, frameworks like ours that demystify large datasets can be instrumental in understanding and mitigating bias. Indeed, we show that finetuning existing classification heads on low spuriosity images is an efficient and effective model-agnostic way to close spurious gaps at little cost of validation accuracy, resulting in fairer, more stable treatment of inputs, regardless of spuriosity.

We also observe surprising instances of negative spurious gaps, i.e. classes where models perform worse when spurious cues are present. These cases are due to spurious feature collision, where one class' spurious feature is also correlated more strongly with another class. Existing benchmarks [73] lack the scale of ours to reveal this atypical yet harmful consequence of spurious feature reliance. Closer inspection reveals high spuriosity images for classes with negative spurious gaps are often mislabeled or contain multiple objects. Thus, spuriosity rankings can help in flagging label noise, and in some cases (via cropping to isolate regions activating core

[1] `salient-imagenet.cs.umd.edu`

neural features), resolving it.

In summary, we present a simple, extensible method for (i) discovering spurious features (ii) measuring and (iii) mitigating model biases caused by these spurious features. Centering on the fact that spurious correlations are determined by data, our solution focuses on better understanding and using the data one already has, in contrast to data-agnostic approaches which focus on altering model training [71, 72]. Arguably, our method provides more practical robustness, as we first interpret potential biases a model trained on the given data is likely to suffer, and then devise solutions specifically tailored to these biases. By shifting focus from model training to data understanding, spuriosity rankings allow for efficient and interpretable robustness interventions, towards more reliable AI.

9.2 Review of Literature

Spurious correlation benchmarks typically consider a small set of predefined spurious attributes fixed across classes [40, 83, 252], and real-world benchmarks tend to be very specific to a narrow task [73]. In both cases, the fixed spurious attributes are annotated manually. Similarly, many have proposed variants of ImageNet's validation set, collecting new data (e.g. with altered renditions, as sketches, in household settings and odd poses, etc [37, 84, 114]) and inspecting the drop in accuracy compared to the original validation set. In contrast, our framework provides a *general* method to uncover and automatically annotate (i.e. per sample) the presence of spurious features relevant to *any* task given its training data, as well as measure and mitigate model bias caused by these spurious features. Since bias arises from data, we believe our step of first interpreting the correlations within a dataset, instead of simply choosing spurious features of

interest apriori, is paramount in order to build *practical* robustness (i.e. to distribution shifts that are naturally occurring and potentially harmful, since they break spurious correlations a model trained on the given data likely uses). Also, we account for the fact that whether or not a feature is spurious depends crucially on its class: e.g. while the *water* feature is spurious for the *otter* class, it is not spurious for a class like *lakeshore* (also, see figure 9.3). Most benchmarks overlook this, defining spurious features class-agnostically. Other approaches to benchmark spurious feature reliance require synthetic data [39, 71, 118] or corrupting spurious regions, which undesirably takes models out of distribution [15, 68]. Instead, our benchmark operates only on natural images one already has, using spuriosity to curate subsets with spurious cues strongly present and absent.

Natural image-based interpretability consists of explaining inferences using examples from the corpus [253, 254] or generated examples intended to visually reflect a traversal between concepts [255]. Similarly, our rankings traverse from sstrong spuriosity to minority examples with spurious cues absent, though we do not require generative models. Other related approaches are influence functions [41] and data models [256], though they are computationally expensive and the former can be fragile [45].

Mitigating spurious correlation reliance often involves altering the training objective to learn more invariant features [71, 80] or directly optimize worst group accuracy [40, 72, 78, 121, 238]. Most methods require extra supervision with respect to non-class-label attributes [257]. Retraining approaches either focus on correcting errors [79, 239, 258] or using balanced data (i.e. dominant spurious cues balanced with minority samples) [76]; our rankings would directly assist in constructing such balanced subsets.

Neural features have been studied to interpret models [49, 259]. Notably, adversarially robust neural features have enhanced interpretability [48, 260]. Singla and Feizi [1] introduced a



Figure 9.3: Examples of the visualizations we use to identify the focus of a robust neural feature. Here, feature 595 focuses on flowers, making it core for *Daisy* and spurious for *Monarch Butterfly*.

framework that uses robust neural features to discover core and spurious data patterns with minimal human supervision. We apply this framework to scalably compute spuriosity rankings. In the next section, we explain this framework and our novel contributions to it, including how to use the framework without requiring its most prohibitive step of adversarial training, thus dramatically increasing its use cases.

9.3 Discovering Spurious Features in ImageNet and Beyond

A key component of our framework for interpreting and improving a model's robustness to spurious correlations is that we first discover relevant spurious features *based on the training data*, as opposed to data-agnostic approaches. Specifically, we leverage the feature discovery method of [1], performing it at an unprecedented scale (i.e. across all of ImageNet). We now provide a brief overview of the method, details on our expansion, and a novel extension of the method with significant impacts.

9.3.1 5000 Reasons Deep Models Use to Perform ImageNet Classification

Despite its ubiquity, ImageNet (and any other large-scale dataset) is opaque in the sense that a human cannot anticipate the patterns a model trained on it will associate with each class.

The reason for this is simple: humans cannot process a million (or even 1000) images at once. Moreover, the patterns a human may use will not necessarily align with those a model will use [261]. Nonetheless, understanding the features (especially the spurious ones) that a model will rely upon is instrumental in anticipating and mitigating the biases a model will suffer from.

To understand the features any general model may rely upon, we inspect the *neural* features of a single, interpretable model; namely, an *adversarially trained* one. Adversarial training leads to perceptually aligned gradients [126], which greatly improve the utility of gradient-based interpretations. Specifically, using the gradient of a neural feature's activation w.r.t. the input image, one can reliably generate a heatmap highlighting the input regions activating the neural feature, and even perturb the input image to visually amplify the cue that the feature detects; the latter method is called a feature attack. Thus, given a robust neural feature, a human can annotate its function as core or spurious for a class by inspecting the images within the class that activate the feature most (we use top 5), along with heatmaps and feature attacks for those images (see Figure 9.3). Note that given a class, one can automatically select important neural features based on the average contribution of the feature to the class logit, which can easily be computed by inspecting feature activations and linear classification head weights. These steps make up the feature discovery framework of [1]; to summarize, (i) adversarially train a model, (ii) automatically select important neural features per class, and (iii) use complementary visualization techniques to annotate a neural feature as core or spurious with minimal human supervision.

Singla and Feizi [1] annotated the 5 most relevant neural features for 232 classes of ImageNet. Through a large-scale human study, we expand the analysis to all 1000 classes, resulting in 5000 annotated class-feature pairs, of which 630 are spurious over 357 classes. We host a web-UI to view all 5000 pairs, offering a direct visual look into the patterns a neural network

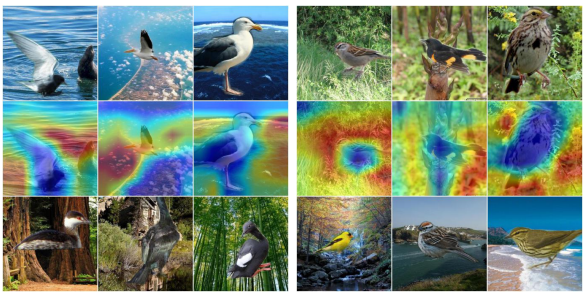
sees and uses across ImageNet. We also verify that heatmaps for annotated features localize the same cue in images whose activation on the feature is in the top 20^{th} percentile for the class, successfully validating 95.3% of annotated class-feature pairs. Thus, we generate $325,000$ feature soft segmentations across ImageNet as a bonus. Crucially, the validation confirms that sorting by feature activation is effective in gathering instances where a feature is present.

While we use the feature discovery method of [1] directly, we make a number of impactful contributions atop it. Namely, we expand its use cases significantly by removing the requirement of adversarial training. Also, we overhaul the original procedure for assessing model reliance on spurious features, making it far more efficient, less biased, and more stable. Key to our improvements is the use of *lowest* activating images as natural counterfactuals to better interpret spurious features and measure their effects, enabling new cross-class and cross-model analyses. Lowest activating images (never used in [1]) are also crucial for computing and closing spurious gaps (Sections 9.4.2 and 9.4.3).

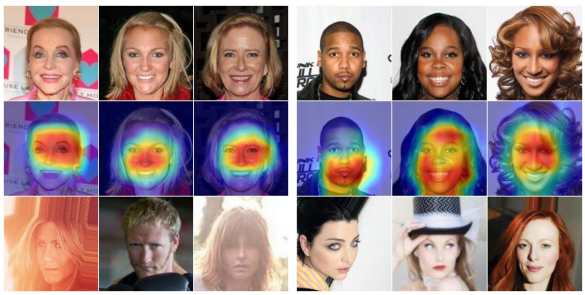
9.3.2 Spurious Feature Discovery *without* Adversarial Training

Adversarially robust neural features are at the heart of our framework, though adversarial training can be challenging, particularly when data is limited. We propose to simply fit a linear layer for a new task atop fixed features from an adversarially robust feature encoder pretrained on ImageNet; prior work shows transfer learning on robust features is very effective [105]. We then perform the usual process of identifying important robust neural features per class and annotating them as core or spurious.

We demonstrate this light-weight extension of the feature discovery framework of [1] on



(a) **Background bias** in Waterbird vs. Landbird Classification; the focus should be on the bird, not the background.



(b) **Racial bias** in Celeb-A Blond vs. Brown Hair Classification; the focus should be on the hair, not the skin.

Figure 9.4: With only linear layer retraining, we detect designated spurious attributes like background in the Waterbirds benchmark (left), as well as new spurious dependencies such as racial bias in Celeb-A Hair Classification (right). Presented above are highest activating images (top row), corresponding heatmaps (middle), and lowest activating images (bottom) for spurious features observed for the classes *waterbird, landbird, blond hair,* and *brown hair* (left to right).

two spurious correlation benchmarks: Waterbirds, and Celeb-A Hair Color classification. Figure 9.4 shows some of the discovered cues. For Waterbirds, we find neural features that focus entirely on backgrounds (the intended spurious feature for this benchmark). The lowest-ranked images within a class for these features contain the opposite background (land instead of water), thus revealing the minority subgroup (waterbirds on land). For Celeb-A, we observe features that spuriously focus on faces (instead of hair). Further, race appears to be homogeneous amongst the top-ranked images, and opposite in the bottom-ranked images. Thus, a model may spuriously associate brown skin with the brown hair class, likely leading to failures for blond-haired brown-skinned people. To our knowledge, *we are the first to uncover this racial bias in the Celeb-A benchmark*, whose intended spurious feature was gender.

To showcase the ease-of-use of our method and the wide-reaching nature of the spurious correlation problem, we apply the lightweight feature discovery framework to a *non*-spurious correlation benchmark, finding numerous spurious cues. We take UTKFace [262], a dataset of face images, and focus on the tasks of gender and race classification. Figure 9.5 visualizes three

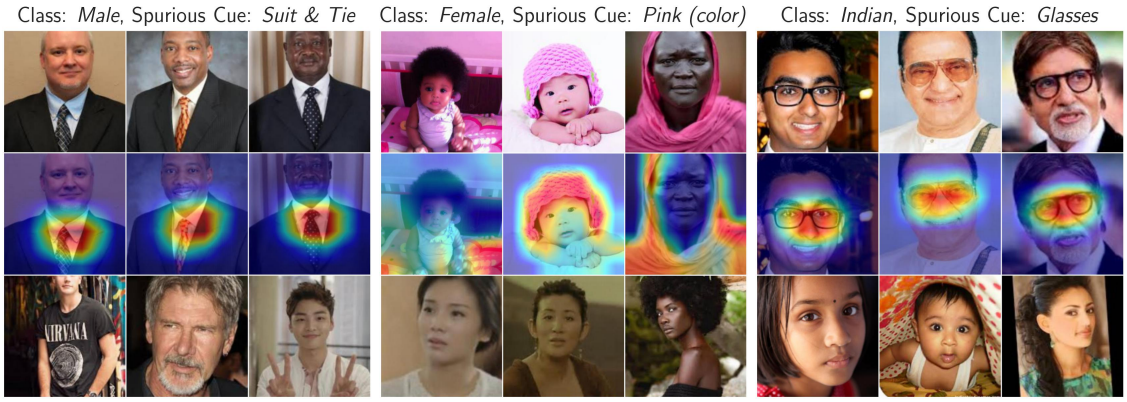


Figure 9.5: Our lightweight extension of [1] also reveals spurious features in *non*-spurious correlation benchmarks, such as UTKFace gender and race classification. Spurious cues are absent in the lowest activating images (bottom row). Thus, sorting by feature activation uncovers minority subpopulations.

detected spurious features: suit and tie for the *male* class, the color pink for the *female* class, and glasses for the *Indian* class. Thus, using robust neural features to discover spurious features is effective, even when the neural features were not trained on the data of interest; simply fitting a linear layer (which can be done in minutes) over these fixed features sufficed in revealing spurious features. The spurious features are prominently displayed in the top activating images, highlighted in the heatmaps, and *notably absent in the least activating images*, indicating that sorting images based on robust neural feature activations can reveal minority subpopulations where spurious correlations are broken. These low spuriosity images prove to be instrumental in measuring and mitigating biases (Section 9.4).

9.3.3 Is a Human in the Loop Necessary?

Our framework involves a human in the loop, who is given (i) concise insight to the cues a model trained on the given data may rely upon and (ii) agency to decide which of these cues model performance should be invariant to. We believe this increased transparency fosters greater

trust with practitioner. Further, the human involvement is limited to viewing a handful of images per class, making our framework tractable (as we demonstrate by performing it on the 1000 class ImageNet dataset). Nonetheless, in cases where minimizing costs is preferred, the human in the loop can be removed, either by automating the feature interpretation step, or by using open-vocabulary models to directly embed concepts into representation space from text descriptions [13, 263]. The key idea of our work is to leverage the interpretability of certain existing models to uncover vectors in representation space corresponding to spurious cues, with which we can scalably sort data by computing similarity to these vectors. The method we present uses axis-aligned vectors of adversarially trained network (i.e. robust neural features), though our framework is not restricted to this instance. We now show how sorting enables for greater utilization of the data one already has, towards resolving the biases caused by relying on spurious correlations.

9.4 Measuring and Mitigating Biases with Spuriosity Rankings

9.4.1 Spuriosity Rankings: Organizing Image Data via Robust Neural Features

Equipped with a method to identify robust neural features that detect relevant spurious cues, we now utilize these features to scalably quantify *spuriosity* (i.e. how strongly spurious cues are present in an image). Letting $\mathbf{r}_i(x)$ denote the activation of image x on the i^{th} robust neural feature, and $\mathcal{S}(c)$ denote the set of neural features annotated as spurious for class c, we compute spuriosity as follows:

*The **Spuriosity** of an image x for class c, with μ_{ic} and σ_{ic} denoting the mean and standard deviation of activations on feature i over class c, is approximated as* $\frac{1}{|\mathcal{S}(c)|}\sum_{i\in\mathcal{S}(c)}\frac{\mathbf{r}_i(x)-\mu_{ic}}{\sigma_{ic}}$.

Essentially, we use the activation of robust neural features as spurious concept detectors to

proxy the degree to which relevant spurious cues are present in each image of a dataset *efficiently*; that is, with only a single forward pass of the dataset through the adversarially trained network. Given a class, averaging the distribution-adjusted activations of relevant spurious neural features yields a single scalar value for each image, with which images can be ranked within their classes.

Note that we define the spuriosity of an image *for a specific class*, since whether a neural feature is relevant or spurious depends on what class it is being used for: figure 9.3 shows an example feature (flowers) that is core for one class (*Daisy*) and spurious for another (*Monarch Butterfly*). Accounting for the crucial class-dependence of the spurious correlation problem is unique to our framework.

9.4.2 Measuring Bias: Computing Spurious Gaps

Relying on spurious cues can bias a model, as its performance may degrade when spurious correlations are broken. Measuring this bias can be expensive, as it requires collecting new data where spurious cues are absent. With spuriosity rankings, we instead can easily select such a subset *from one's own data*. Indeed, high spuriosity images prominently display spurious cues, while low spuriosity images do not (Figure 9.1). Thus, a natural metric enabled by spuriosity rankings is **spurious gap**, defined as the drop in accuracy between the top-k highest and lowest spuriosity validation images per class. We evaluate spurious gaps over the 357 ImageNet classes with at least one spurious feature discovered.

Figure 9.6 (left) shows accuracy on the $k = 10$ lowest (y-axis) vs. highest (x-axis) spuriosity images for 89 diverse models pretrained on ImageNet [2]. We observe *all* models to be biased,

[2]except for the CLIP models, which are trained on more data, and also, perhaps not surprisingly, happen to be the most exceptional in their robustness.

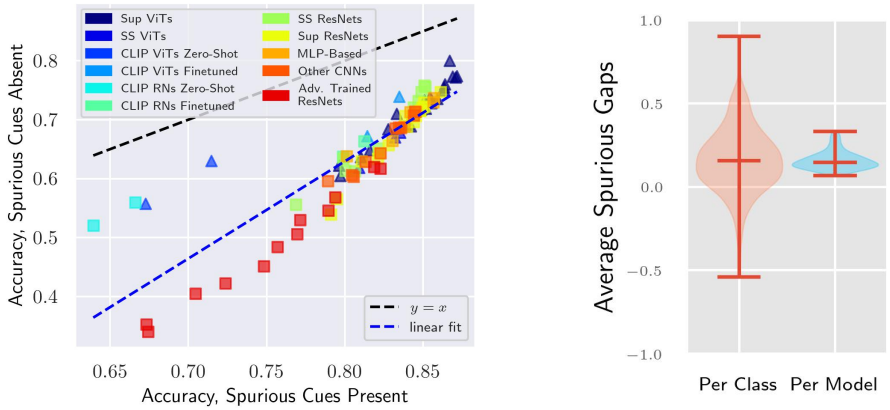


Figure 9.6: All models are biased: accuracy is 10 to 30% lower when spurious cues are absent compared to when they are present (**left**). However, bias varies far more significantly across classes than across models (**right**), emphasizing that bias due to relying on spurious cues is crucially class-dependent.

suffering lower accuracy when spurious cues are absent. As in [264], we observe a strong correlation between the two accuracy measures. However, some models stray from this linear trend, exhibiting 'effective robustness' by performing significantly better on images with spurious cues absent than the linear trend predicts (e.g. zero-shot CLIP) or the opposite (e.g. adversarially trained models). Prior work on distributional robustness corroborates our findings on the effective robustness (and lack thereof) of CLIP and adversarially trained models respectively [13, 91], thus validating our metric. Essentially, **spuriosity rankings organize our data so that we can *simulate distribution shifts* caused by breaking spurious correlations, so to assess a model's robustness without needing to collect new data**.

We now turn our attention to class-wise spurious gaps (i.e. we average spurious gaps per class over all models, instead of averaging over all classes per model, as done in Figure 9.6, left). We find that spurious gap varies significantly more across classes than across models (Figure 9.6, right). Specifically, the sample variance is $3.1\times$ larger for spurious gaps across classes than across models, suggesting that the spurious correlation problem pertains more to the data a model

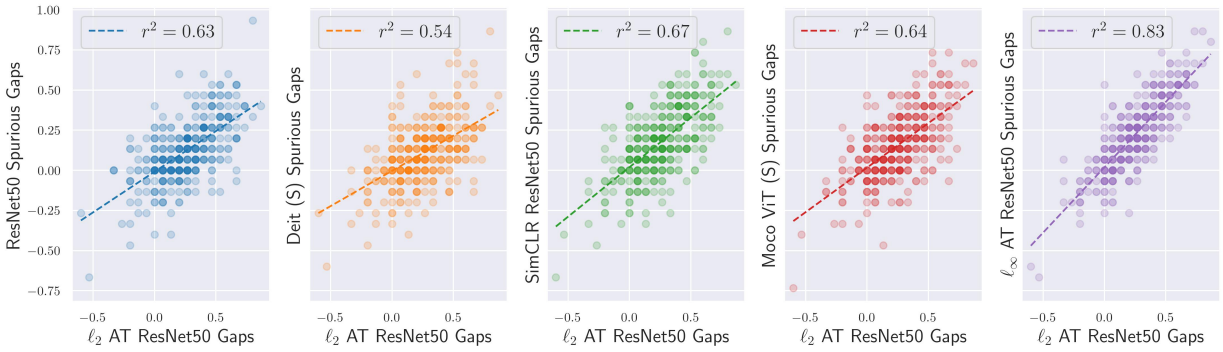


Figure 9.7: Class-wise spurious gaps are strongly correlated across models. While spuriosity rankings are empirically computed with the help of a single ℓ_2 adversarially trained (AT) ResNet, diverse models show similar sensitivities to the detected spurious features.

is trained on than to the specifics of the model itself. That is, we argue that class-wise spurious gaps vary so much because each class has its own set of spurious cues that are correlated with the class label to different degrees. In contrast, despite differences in training procedure and architecture, **all models learn from the same data, and thus, absorb the same biases**. To investigate this claim, for each pair of models in our study, we compute the correlation between class-wise spurious gaps. On average, we observe a high Pearson's r of 0.69.

Figure 9.7 visualizes this strong correlation for five diverse models compared to the interpretable model used to discover spurious features in Section 9.3. We highlight this case as it justifies our approach in using a single model (namely, an ℓ_2 adversarially trained ResNet50) to reveal biases for any general model. Because the spurious correlations a model learns depends far more on *what* it is trained on (i.e. the data) than *how* it is trained, we can generalize the spurious features we discover with one model to others. The results of our large-scale empirical analysis support this claim, as all models display bias to the spurious cues discovered with our interpretable model, and furthermore, the degree to which these models are biased per-class are strongly correlated. While it is not surprising that data determines the spurious features a

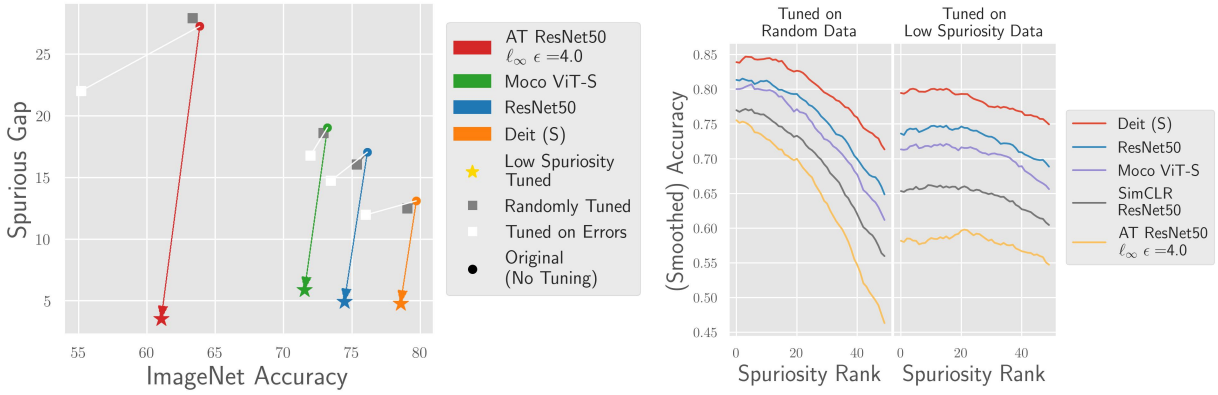


Figure 9.8: (**Left**) Finetuning on low spuriosity images closes reduces spurious gap to below 5% at little cost of validation accuracy. (**Right**) Closing spurious gap removes bias to images with high spurious ranks; that is, regardless of spuriosity, the model performs with roughly the same accuracy.

model learns to rely on, we stress this point, as many existing methods for spurious correlation robustness are data-agnostic. In contrast, our method is *data-centric*. Namely, we first discover the spurious features models may rely upon *based on the data*, via an interpretable (essentially, a surrogate) model trained or fit on the data. Then, with spuriosity rankings, we organize our data to allow for efficient measurement of biases per class. We now show how wisely selecting data, via our rankings, for further tuning can mitigate model bias caused by spurious features.

9.4.3 Mitigating Bias: Closing Spurious Gaps

Having identified a bias across ImageNet-trained models, where images that lack spurious cues common to their class are predicted with considerably lower accuracy, we now present a very simple and efficient manner to mitigate this bias at little cost of validation accuracy. In general, to improve some pretrained models on a more specific downstream data distribution, one simply finetunes the model on new data from the target distribution. With spuriosity rankings, we remove the need to acquire new data, as we instead simply select from the training data we

already have. In this case, we seek to improve the performance of models on images with spurious cues absent; *this is precisely the low spuriosity images*. Applying our rankings to training data, we can extract a small subset that captures minority subpopulations under-represented in the overall training distribution. Then, we tune the existing classification head on this small subset so to recalibrate the model towards reduced reliance on spurious features and fairer treatment of samples, regardless of spuriosity.

We choose to only modify the classification head, as prior work suggests deep models already learn features that are sufficiently informative to correctly predict minority subpopulations [76]. Also, this way, we minimally change the existing model, leading to better retention of validation accuracy, and *extremely fast* bias mitigation, as we only optimize a tiny fraction of the total model parameters using a small fraction of the entire dataset. Specifically, we tune the classification heads of five models spanning diverse architectures (ResNet and ViT) and training procedures (supervised, self-supervised, adversarial) on the 100 lowest spuriosity images for each of the 357 classes for which we discover one spurious feature. This tuning occurs *in minutes* on a single GPU, especially since the data passes through the entire model only once; after caching features, all computation is limited to the linear classification head. Also, we employ early stopping, halting tuning once spurious gap drops below 5%, so to avoid overfitting to the minority subpopulations at the cost of overall accuracy.

Figure 9.8 visualizes the results of our bias mitigation, plotting the spurious gap (over all 357 classes) vs. validation accuracy before and after our tuning. We also include as baselines (i) tuning on a subset consisting of 100 randomly selected images from each of the same 357 classes and (ii) tuning on misclassified inputs from each class, inspired by existing spurious correlation mitigation techniques [79, 239, 258]. We find that **low spuriosity tuning mitigates bias**

at a minimal cost to validation accuracy, while the baselines do not. In fact, tuning on errors only slightly closes spuriosity gaps, while causing a more significant drop in overall accuracy. Namely, low spuriosity tuning closes spurious gaps, reducing it by 10 to 20%, while only sacrificing 1 to 3% validation accuracy. In the subplot on the right, we see how low spuriosity tuning effectively removes bias to high spuriosity images. While the randomly tuned baseline (like the original model) has far better accuracy on images with high spuriosity ranks, the low spuriosity tuned model has far more stable treatment of all samples, evidenced by a much flatter curve when plotting accuracy vs. spuriosity rank.

9.5 Additional Application: Flagging and Fixing Labeling Errors

We now demonstrate an additional manner in which spuriosity rankings can help manage and improve the data one already has. Modern datasets, like models, are opaque in the sense that it can be challenging to answer simple yet significant questions about them, largely due to their ever-increasing size. One such question is what spurious features a model will learn to rely upon, and to what degree, after training on a given dataset; our spuriosity rankings framework is designed to answer this question. A related question is how accurate the labels in a dataset are. Even in some of the most widely used vision datasets like ImageNet, label noise has been observed to be pervasive [265], and when it is resolved, models trained on the refined data experience improved accuracy and robustness [266].

A common culprit for label noise in ImageNet is the presence of multiple objects in the same image. Similarly, co-occurring objects are amongst the most common types of spurious cues we observe models to rely upon. For example, car wheels almost always co-occur with a

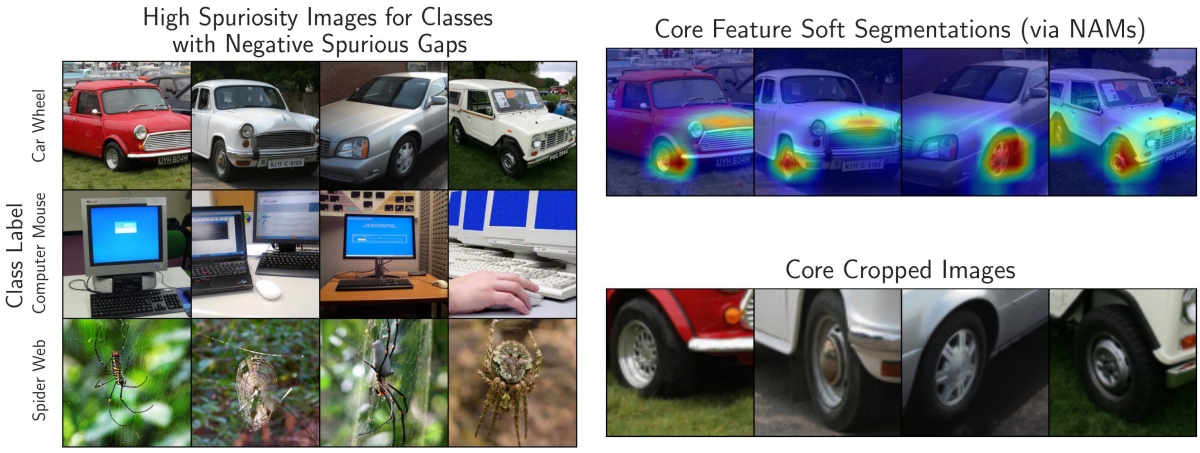


Figure 9.9: (**Left**) Training images with highest spuriosity for the classes with the most negative spurious gaps. These images often contain multiple objects, at times more prominently showing an object from a class different than the image's label (e.g. *car*, *monitor* or *keyboard*, *spider*). (**Right**) Using neural activation maps for neural features annotated as *core* (i.e. not spurious) for the class may serve as an automated way to crop out the conflicting objects.

car, and so, a model may associate the metallic body of a car with the *car wheel* class. Indeed, we discover this spurious feature in our analysis. However, while we typically expect a model to underperform when a spurious cue is absent, we observe the opposite: the car wheel class has a *negative* spurious gap, with models on average having 54.2% lower accuracy on high spuriosity car wheel images than low spuriosity ones. While uncommon, negative spurious gaps are still prevalent, occurring in 15.7% of classes we study. This highlights a second, often overlooked, harm incurred by spurious feature reliance. Namely, since spurious features are not essential to a class, they can be correlated with multiple classes at once, leading to *spurious feature collision*. Indeed, a spurious feature for one class may also be core for another, like the car body being spurious for the *car wheel* class and core for any of the numerous car classes. This is actually the case for a majority of the spurious features we discover, with a staggering 63.8% of them also being core for a different class. In these instances, if a feature f is correlated more with class c_1 than c_2, high spuriosity images from c_2 may be misclassified to c_1.

Taking a closer look at classes with severely negative spurious gaps reveals instances of mislabeled training data. Figure 9.9 shows high spuriosity images for classes with the most negative spurious gaps. These images often more prominently display co-occurring objects different from the class label (such as a car, monitor, or spider) which themselves pertain to another class. Thus, one can automatically flag potential instances of label noise by inspecting high spuriosity images for classes with negative spurious gaps. To quantitatively validate this claim, we leverage ImageNet ReaL labels [265], which indicate for each ImageNet validation image whether objects from multiple classes are present. ReaL labels show 14% of validation images contain multiple objects. We find that for classes with the 5 most negative spurious gaps (averaged over our model suite), high spuriosity (i.e. ranked in top 10% of spuriosity) validation images contain multiple objects in 80% of cases. In contrast, low (bottom 10%) spuriosity images from the same classes contain multiple objects in only 8% of cases, indicating that the label noise is likely a result of the hypothesized spurious feature collision. Similarly, for classes who's spurious gap is less than -20%, ReaL labels reveal 42% of the high spuriosity images to contain multiple objects, many times higher than the average rate of label noise.

Taking label-noise flagging a step further, in some cases, we can utilize the neural activation maps of core robust neural features for these classes to automatically refine the mislabeled high spuriosity images. Namely, we can crop the image based on the region that activates core features for the class, to focus on the actual object of interest and remove the spurious co-occurring one. Alternatively, segmentation models may be used to refine flagged samples.

9.6 Conclusion

In this work, we tackle the problem of spurious correlations in image classifiers by shifting attention from training algorithms to *data*. As datasets continue to grow, it can be challenging to answer even the simplest questions about what resides within them. We propose to leverage interpretability tools to harness machine perception towards scalably organizing data. Namely, we sort data within each class based on spuriosity, as measured by neural features in an interpretable network. With spuriosity rankings, we can easily retrieve natural examples and counterfactuals of relevant spurious cues from large datasets, enabling efficient measurement and mitigation of spurious feature reliance. We demonstrate the feasibility of our approach on a vast set of models on ImageNet – a far larger and more realistic setting than most prior spurious correlation benchmarks. Further, we observe highly similar biases across our diverse model set, suggesting that biases may be determined more so by what a model sees than how it is trained. We hope our work spurs further exploration into how to best understand and utilize the data we already have, towards more robust and interpretable models.

Part III: Applications to Sociotechnical Challenges

Thus far, we have observed how standard evaluation procedures may not suffice to uncover corner cases that give rise to reliability issues. However, promisingly, we find that models can be used in auxiliary fashion to aid in more scalably searching the space of potential limitations, as well devising mitigation strategies for the uncovered failure modes. Now, we turn our attention specifically to limitations that have direct social impact. Instead of studying distribution shifts that break spurious correlations and lead to errors like misclassifying a bird in the absence of branches, we look at naturally occurring distribution shifts – e.g. geographic ones – which lead to issues like image classifiers achieving far lower accuracy for images from non-western countries. Specifically, we present a benchmark for obtaining a per-country metric for LLM factuality, towards uncovering geographic disparities. Then, we investigate this same disparity in the context of zero-shot classification with vision-language models. In this case, we posit that the adoption of the prevailing paradigm for classification leads to an under-utilization of the recognition capabilities present in modern VLMs. We then propose a new classification paradigm, leveraging an LLM as an auxiliary component, that leads to reduced disparities and greater transparency, once again showing the power and potential of building AI systems where individual black-box components collaborate in a more transparent manner. Finally, we tackle the task of copyright protection of artistic styles. This is a fundamentally interdisciplinary challenge, relevant to art, legal, and technical communities, that has direct human and material consequences. Moreover, transparency is paramount in this setting, as the ultimate arbiters of copyright infringement are people, who will need to be able to interpret how an automated tool arrives at its measurement of stylistic similarity.

Chapter 10: WORLDBENCH: Quantifying Geographic Disparities in LLM Factual Recall

This chapter details work from a FAccT '24 paper co-authored with Elham Tabassi (my mom!) and Soheil Feizi [210]. While we have studied distribution shifts in an academic sense, we now take a look at a very real distribution shift that can have adverse societal impact if overlooked: *geographic* shift.

As large language models (LLMs) continue to improve and gain popularity, some may use the models to recall facts, despite well documented limitations with LLM factuality. Towards ensuring that models work reliably *for all*, we seek to uncover if geographic disparities emerge when asking an LLM the same question about different countries. To this end, we present WORLDBENCH, a dynamic and flexible benchmark composed of per-country data from the World Bank. In extensive experiments on state of the art open and closed source models, including GPT-4, Gemini, Llama-2, and Vicuna, to name a few, we find significant biases based on region and income level. For example, error rates are 1.5 times higher for countries from Sub-Saharan Africa compared to North American countries. We observe these disparities to be consistent over 20 LLMs and 11 individual World Bank indicators (i.e. specific statistics, such as population or CO_2 emissions). WORLDBENCH also enables automatic detection of citation hallucination, where models cite the World Bank itself while providing false statistics, and a manner to assess when

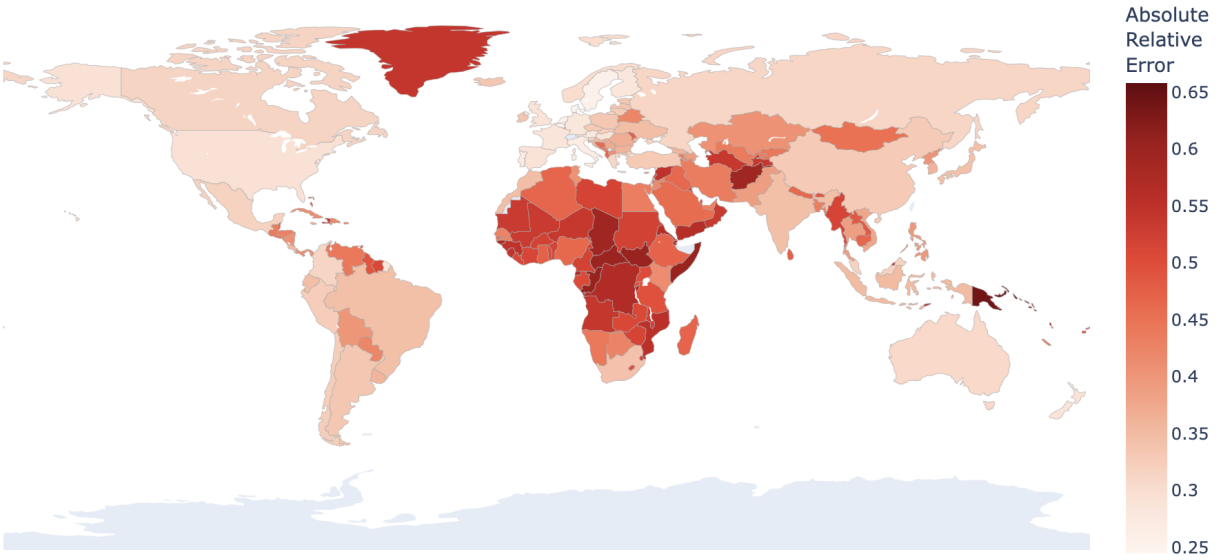


Figure 10.1: WORLDBENCH leverages World Bank data to assess the ability of large language models (LLMs) to recall factual information about specific countries. Above, we plot the absolute relative error per country, averaged over 11 global development statistics queried to 20 state of the art open source and private LLMs. WORLDBENCH reveals significant geographic disparities in LLM factual recall.

an LLM's stored facts begin to go out of date. We hope our benchmark will draw attention to geographic disparities in existing LLMs and facilitate the remedying of these biases.

10.1 Introduction

Large language models (LLM) exhibit remarkable performance on a wide array of tasks, from summarizing the news to writing code to answering trivia questions [267, 268, 269]. Impressively, LLMs have also been effective on real-world benchmarks. For example, GPT-4 [270] has been shown to pass the licensing exams for both legal [271] and medical professions [272, 273]. However, LLMs are also known to hallucinate, where they generate inaccurate text in a plausible manner [274]. This can pose particular risks for factual recall tasks. Given the black box nature of LLMs, continued development and application of diverse benchmarks is instrumental in

understanding when LLMs can be trusted to answer reliably.

In addition to issues with correctness, AI in general has well documented challenges with performance disparities, in which seemingly strong models fail more frequently for some subset of inputs than others. Performance disparities can manifest as fairness issues when the subset of inputs where the model underperforms is characterized by sharing a sensitive attribute. For example, Buolamwini and Gebru [9] identified widespread performance disparities along race and gender lines across commercial facial recognition systems, while others have shown that object recognition models suffer performance drops when images originate in lower income countries [275, 276]. Similarly, Ojo et al. [277] show LLMs are less performant when tasks are posed using African languages instead of English. A key first step to building models that work *for all* is creating benchmarks to quantify not only performance, but also performance disparities.

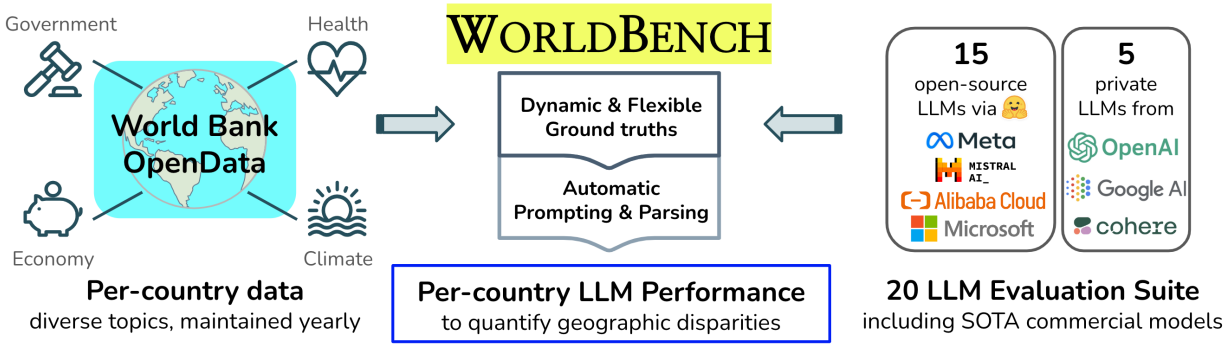


Figure 10.2: Overview of WORLDBENCH. Our benchmark provides a manner to quantify the performance of large language models (LLMs) on a per-country basis. We disentangle data collection from evaluation by utilizing the World Bank's data bank, which contains statistics (called indicators) pertaining to numerous diverse aspects of global development. Crucially, the data is available for nearly all countries and is updated year to year. With WORLDBENCH, one can flexibly select specific statistics of interest, and dynamically re-evaluate models as time passes to see if they remain up to date. In this work, we uncover substantial geographic disparities in LLM performance for a wide range of models released by industry leaders, revealing the inequities pervasive across state of the art LLMs.

To this end, in this work, we introduce a novel benchmark called WORLDBENCH to un-

cover if *geographic* disparities emerge in LLM factual recall. In other words, we ask, *are LLMs more accurate in answering questions about some parts of the world than others?* To systematically tackle this question, we compute LLM performance on a country-wise level, by way of utilizing per-country indicators (i.e. statistics) from the World Bank [278]. We build and validate (via human inspection) an automated, indicator-agnostic prompting and parsing pipeline to interface with the World Bank data, summarized in Figure 10.2. This way, any set of indicators can be used in future variations of WORLDBENCH, without having to change our code, which we will make public. In our study, we incorporate 11 diverse indicators, each having data for about 200 countries, resulting in a total of $2,225$ questions per LLM.

Indicator	Metric
Population	Total Population
Unemployment	Unemployment As A Percent Of The Total Labor Force
Maternal Mortality Rate	Maternal Mortality Ratio As Number Of Deaths Per 100,000 Live Births
Women In Parliament	Proportion Of Seats Held By Women In National Parliaments (As A Percent)
Education Expenditure	Government Expenditure On Education As A Total Percent Of Gdp
Electricity Access	Percent Of The Total Population That Has Access To Electricity
Agricultural Land Percent	Percent Of Total Land Area That Is Agricultural
CO_2 Emissions	Amount Of Carbon Dioxide Emissions In Metric Tonnes Per Capita
GDP	Gdp Measured In Us Dollars
GDP PPP Per Person Employed	Gdp At Purchasing Power Parity (Ppp) Per Person Employed
Renewable Energy Ratio	Renewable Energy Consumption As A Percent Of Total Final Energy Consumption

Table 10.1: Global development indicators in WORLDBENCH, each defined and maintained by the World Bank.

We evaluate 20 state of the art LLMs released in 2023, ranging from open-source models like Llama-2 and Vicuna [279, 280], to private commercial ones accessible via API, including GPT-4 and Gemini [270, 281]. As visualized in Figure 10.1, when averaging over all LLMs and indicators, we observe substantial differences in per-country error, with African countries seemingly incurring the largest errors. Using country categorizations defined by the World Bank, we quantify disparities across 7 regions and 4 income groups, finding that LLMs are most accurate

for countries from Western regions and the high income category. Problematically, these error rates rise by a factor of about $1.5\times$ when moving to the region (Sub-Saharan Africa) and income group (low income) for which models are least accurate. Moreover, we find these disparities and their order (i.e. which groups have most/least error) to be consistent when inspecting LLMs or indicators individually. That is, *all* 20 *LLMs exhibit geographic disparities in factual recall*.

In addition to our main result, we utilize the temporal aspect of the World Bank data to conduct extra analyses, such as automatically cross-checking LLM generated "citations" which turn out to be hallucinated, and inspecting error as a function of the groundtruth year, finding that some LLMs in our suite may already be slightly out of date.

In summary, we present WORLDBENCH, a flexible benchmark for understanding LLM factual recall abilities on a per-country basis. With WORLDBENCH, we conduct a large scale evaluation of 20 LLMs, and find pervasive geographic disparities across regions and income levels. We hope our benchmark can facilitate further research on the fairness of LLMs, towards building models that work well *for all*.

10.2 Related Work

Evaluating Factual Recall. Recent works have documented the performance of LLMs in factual recall: [282], [283], [284], [285]. The general conclusion to these works is that while existing LLMs appear capable in answering certain factual question, their factual recall is less than perfect, as models can hallucinate completely fabricated information [274]. Zhang et al. [286] specifically investigated the recall of geographic information, though their study is limited to GPT-4 and does not inspect disparities. Some works (e.g., [282], [284]) linked factual recall

to ‘popularity’, showing that error rate increases for less popular entities. While those studies categorize facts by popularity, each question in our benchmark has an associated country, as well as Region and Income group. These additional annotations enable going beyond overall error, so to assess geographic performance disparities in factuality.

Bias. The issues of bias and fairness in AI are of immense societal impact. Several studies have observed computer vision models to exhibit disparate performance when grouping inputs by race, gender, and across income levels and geographies, for tasks like facial recognition, object classification, and diverse image generation [9, 275, 276, 287]. In the realm of language processing, Ojo et al. [277] observed a performance gap when tasks are presented in African languages. To the best of our knowledge, our study is the first to propose an automated and systematic examination of country-wise disparities in LLM factual recall, which in turn enables inspection of disparities across regions and income groups.

Benchmarks. Other works have noted and sought to improve challenges associated with evaluating factuality, primarily for tasks like summarization, where constructing a similarity metric between generated and reference texts is nontrivial. In our case, we design our benchmark to obtain numeric answers from LLM repsonses, with which we can compare to groundtruth values with the simple metric of absolute relative error. Further, we utilize a reputable third party (the World Bank), so that (i) the questions asked are relevant, (ii) inputs are grouped into salient cateogries, and (iii) groundtruth answers are accurate and up-to-date.

10.3 Methods: WORLDBENCH

10.3.1 Data

Our benchmark is constructed directly from statistics collected and maintained by the World Bank. The World Bank is a global organization with nearly 200 member countries, whose mission is to reduce extreme poverty via sustainable solutions to promote shared prosperity, particularly in developing countries [278]. The World Bank tracks numerous global development **indicators**, from 20 wide ranging categories, such as Climate, Health, and Poverty, to name a few. These statistics are freely available to the public and updated yearly. Importantly, the data are collected *per country*, meaning that regardless of the size, wealth, or location of a country, it is represented in the World Bank's data. We leverage this publicly available open data to build WORLDBENCH, a benchmark to quantify the degree to which language models can recall facts about *all* countries in the world.

Our benchmark offers a few unique advantages to most existing benchmarks. First, and most importantly, WORLDBENCH equitably represents all countries. Thus, we can query a language model for the same exact statistic for completely different countries, enabling direct comparisons across countries to uncover disparities in performance. Next, data quality and licensing is assured, as it comes from a globally reputable source which explicitly allows for its use by the public. Third, our benchmark is dynamic and flexible. The dynamic nature comes from the fact that the statistics are updated on a yearly basis, enabling the longevity of our benchmark, as well as analysis of LLM factual recall along a temporal dimension (see §10.6.2). The flexibility is borne out of the vast number of indicators one could choose from. In other words, if one sought

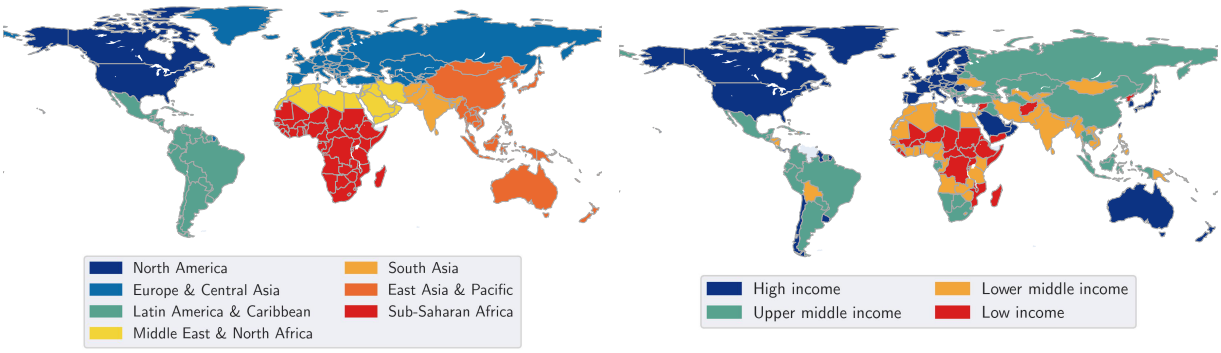


Figure 10.3: World Bank categorizations of countries into 7 regions (left) and 4 income groups (right).

to better examine the ability of language model to recall facts about the environment, they can elect to choose indicators from the Climate category. In contrast, if a language model is being developed for financial purposes, one could focus on indicators from the Economy and Growth categories.

In this study, we select 11 indicators, as shown in Table 10.1. The indicators are chosen to represent multiple different categories, and qualitatively are amongst the indicators that are easier to understand for lay people (i.e. non-experts in global development, like AI researchers). In total, there are $2,225$ questions, reflecting an average of 202 countries with groundtruth data per indicator studied.

Country categorization. The World Bank also provides various categorizations of countries, based on geographic or economic reasons [288]. We focus on two high level categorizations, visualized in Figure 10.3, which divide the world into 7 *Regions* and 4 *Income groups*. We note that, like the collection and maintenance of the groundtruth data for our benchmark, country categorization is carried out by an external body (i.e. the World Bank) to the model producers and evaluators. We hope that the disentanglement of these three parties enables a more objective comparative analysis, informed by experts on global development.

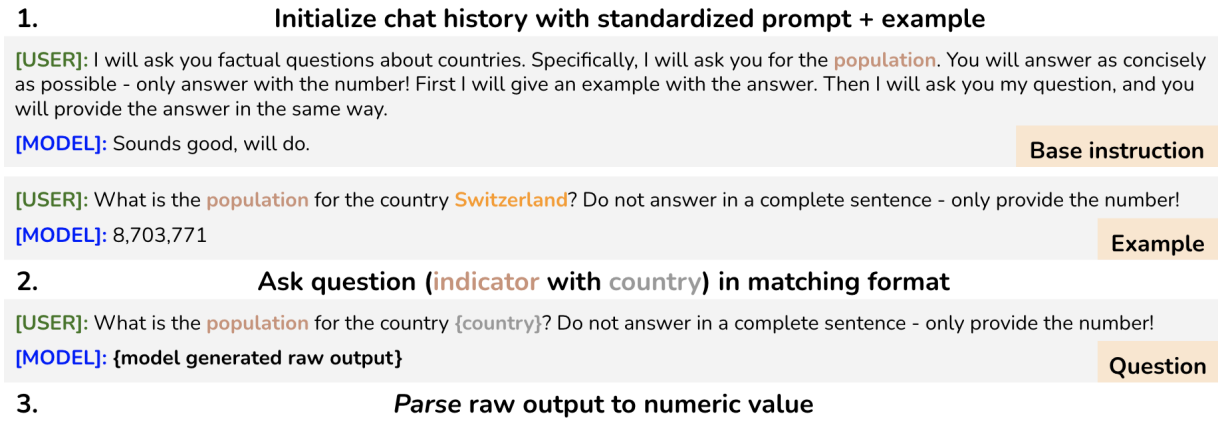


Figure 10.4: Standard pipeline for extracting numeric answers from LLMs. Each question is defined by a query (i.e. Before asking a language model a question, we prompt it with a base instruction and example. Then, we automatically parse the raw output to obtain a numeric value which can be compared to the groundtruth data.

10.3.2 Language Model Evaluation

While the World Bank's open data is crucial to our analysis, additional steps are needed to interface with the available data scalably. To enable large scale evaluation of LLMs, we design a procedure to obtain a numeric answer given an arbitrary indicator, country, and LLM of interest. Namely, we utilize a template prompt to guide models to provide answers in a mostly uniform fashion, and then apply an automated parsing method to extract the numeric value from the raw LLM output. We detail these steps below, as well as results from human studies to validate the correctness of our pipeline. We also explain how we compute errors, given numeric answers from LLMs and the World Bank's groundtruth data.

Prompting. Our standard prompt consists of a base instruction, an example, and a template question filled in with values for the indicator and country of interest. Figure 10.4 displays the base instruction and example. For consistency, we fix our choice of example country, electing Switzerland, as it has groundtruth data for all indicators in our study; we confirm results are sim-

ilar when using alternate example countries. Importantly, we prompt the model to only provide the number in its response. Without this instruction, models generate longer free-from responses, increasing the difficulty of automatically extracting numeric values and the the computational cost of our benchmark. For every question (i.e. combination of an indicator and country), we first initialize the chat history of the LLM of interest with the base instruction and example, and then ask the question. Notably, all three components are modular with respect to the country and indicator of interest, allowing for them to work for any World Bank indicator.

Parsing. Despite the instruction to 'only provide the number', LLMs at times exhibit undesirable, like including other text (e.g. special tokens) or repeating the question with new countries and responding to itself again and again. We design an automated parsing method to scalably extract a numeric value from the raw LLM outputs. The parsing method removes special characters, and in most cases, extracts the first numeric value provided. We also account for special cases like, for example, where a suffix (e.g. 'million' or 'billion') is used. In a small number of cases, the LLM either provides no output, an invalid output (e.g. a number with two decimal points), or abstains from answering. For these outputs and any others where the parsed number cannot be converted to a float, we exclude them from further analysis.

Error metric. To compare numeric values, we utilize absolute relative error, computed as follows: given two scalars a, b, we define *Absolute Relative Error* as $\frac{|a-b|}{\max(a,b)}$. Essentially, this metric conveys by what percent two measures are different from one another. For example, an absolute relative error of 0.1 means that one value was 10% larger or smaller than the other. Notice that absolute relative error always falls between 0 (because all values we encounter are non-negative) and 1 (because the denominator is the maximum of the two positive values). We elect to use relative error over absolute error because the ranges of values varies dramatically

across indicators, with the population indicator having some groundtruth values in the millions and billions, while others (e.g. unemployment) take on values under 10.

Validation. Over 20 LLMs and 11 indicators ($44.5k$ total questions), automated parsing obtains a numeric answer 88.9% of the time. We further validate the correctness and completeness of our pipeline via three manual inspection studies. First, we check 450 random cases where a numeric answer could not be extracted. In 85.2% of cases, the LLM did not provide a parseable answer. Thus, our parsing is mostly complete, as **we obtain a numeric value in 98.2% of cases where an answer can be parsed**. To verify the correctness of the parsing, we first check 945 randomly selected raw LLM outputs where a numeric value was parsed. **In 98.7% of these cases, the parsed value was correct**. Then, we take a closer look at parsed responses that incurred high (over 0.85) absolute relative error compared to the groundtruth value. For 825 randomly selected high error cases, the parsing was manually verified to be correct 93.7% of the time. Motivated by this slightly lower correctness rate, we also analyze median errors over groups, where observed trends are consistent (and disparities over Regions and Income groups are even larger). We conclude that our prompting and parsing pipeline is largely complete and correct. Nonetheless, when evaluating a new LLM, we recommend verifying the parsing behavior using the four validations we outline above, as individual LLMs can have unique idiosyncracies (e.g. special tokens or output patterns) that potentially could affect parsing. Along with all code, we will also publicly release methods to facilitate automatic and manual verification of parsing.

Groundtruth selection. For each indicator and country, data is available over a span of many years, though certain values are missing. To define a single groundtruth value for per country per indicator, we average the statistic over the past three years. The primary motivation for this strategy is to maximize the number of countries included in our study. Alternatively,

one could select a specific year to draw all groundtruths from, though the number of countries considered would be lower than the averaging strategy. In an additional analysis not shown, we compare groundtruth values obtained via different selection methods, and observe groundtruths to only vary by a small amount. We also explore specifying a year when querying LLMs, and observe consistent results with respect to performance disparities to those observed without year specification in the query. Lastly, we more closely inspect overall error rates between LLM responses and groundtruths selected by specifying a year in section 10.6.2, to gain insight on if LLM responses are dated (i.e. more accurate for a prior year than the most recent year).

10.4 Evaluation Suite

We seek to evaluate a wide array of language models, including both open source and private. For the **open source models**, we utilize Huggingface's transformers library [289] to obtain and operate 15 models (and respective tokenizers). Namely, from Meta's LLama-2 [279], we include both base and chat-tuned versions of the $7B$ and $13B$ models, where $7B$ indicates 7 billion parameters. We also include two Vicuna models ($7B$ and $13B$), which are fine-tuned from Llama-2. From Microsoft, we have $7B$ and $13B$ Orca-2 models [195], as well as Phi-2, the smallest model in our suite with just $2.7B$ parameters. From Mistral-AI, we include the $7B$ instruction-tuned model [290]. We also study Zephyr-$7B$ β [291], tuned from a Mistral-AI model. Lastly, we include $7B$ and $14B$ Qwen models from Alibaba Cloud, both with and without chat-tuning [292]. For **closed source models**, we include the following LLMs. From OpenAI, we evaluate gpt-3.5-turbo and gpt-4 [270]. From Google, we evalute Gemini [281]. From Cohere, we evaluate the 'command' model, as well as the same model equipped with retrieval augmented

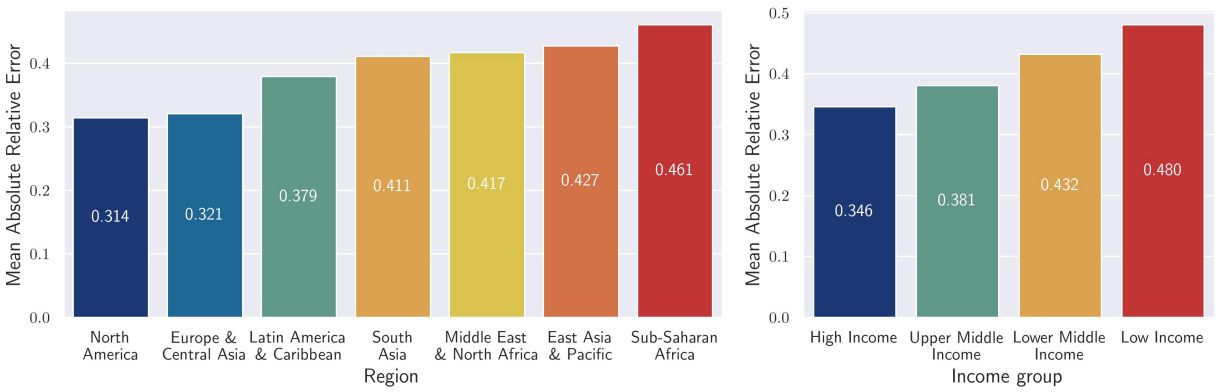


Figure 10.5: Language models exhibit disparate performance for countries from different regions and income groups. Error rates are lower for western and high income countries. Mean absolute relative error rate per region and income group reported over all 11 queries and 20 language models studied. We note that the best performing LLMs have much lower error rates than the averages presented above (see figure 10.7).

generation (RAG) [293]. RAG is a procedure where a langauge model can retrieve relevant documents (in this case, from the internet) and look over them before generating a response.

10.5 Results: Pervasive and Consistent Geographic Disparities

10.5.1 Large disparities across Regions and Income groups

Figure 10.5 visualizes our central finding. Over 20 LLMs and 11 World Bank indicators, we observe substantially disparate average performance based on the Region and Income group of the country of interest. Namely, the mean absolute relative error is 0.316 and 0.321 for countries from North America and Europe & Central Asia respectively. In contrast, the mean absolute relative error rises to 0.461 for countries from Sub-Saharan Africa, which is about $1.5\times$ higher than the error for North America. For Income groups, mean absolute error rises steadily as the income level drops, with the lowest error being 0.346 for high income countries, and the highest error being 0.480 for low income countries.

10.5.2 Error nearly triples between some countries

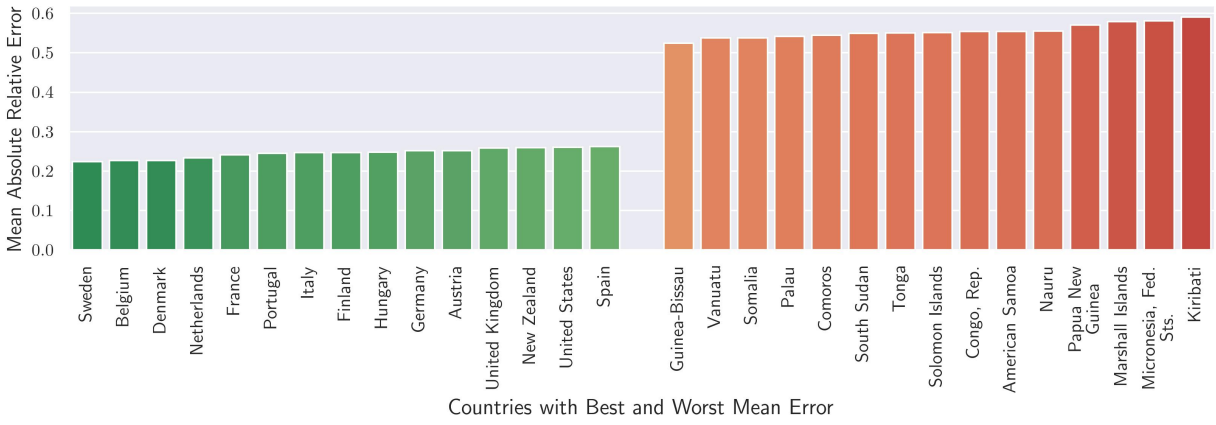


Figure 10.6: Error rates can vary significantly across countries, with some countries experiencing nearly $3\times$ higher absolute relative error than others. Strikingly, all of the 15 countries with the lowest error rates fall in the high income category, while all of the 15 countries with the highest error rates fall in the low income category.

On a per country basis, disparities can become even more pronounced. Figure 10.6 visualizes mean absolute error rate per country for the countries that, when asked about, language models (on average) have the most and least amount of error. We observe that 13 of the 15 countries that incur the least amount of error are European, while all 15 of these countries fall are categorized as high income. On the other hand, countries that incur the most error are all categorized as low income. Strikingly, error rises by a factor of nearly 3 across the two groups.

10.5.3 Consistent disparities across LLMs and indicators

Previously, we presented results averaged over all LLMs and indicators, grouped either by country or category (i.e. Region or Income group). We now inspect performance along the axes of LLMs and indicators separately, starting with LLMs. In addition to absolute relative error, we also employ a second metric to summarize differences in performance across certain categories.

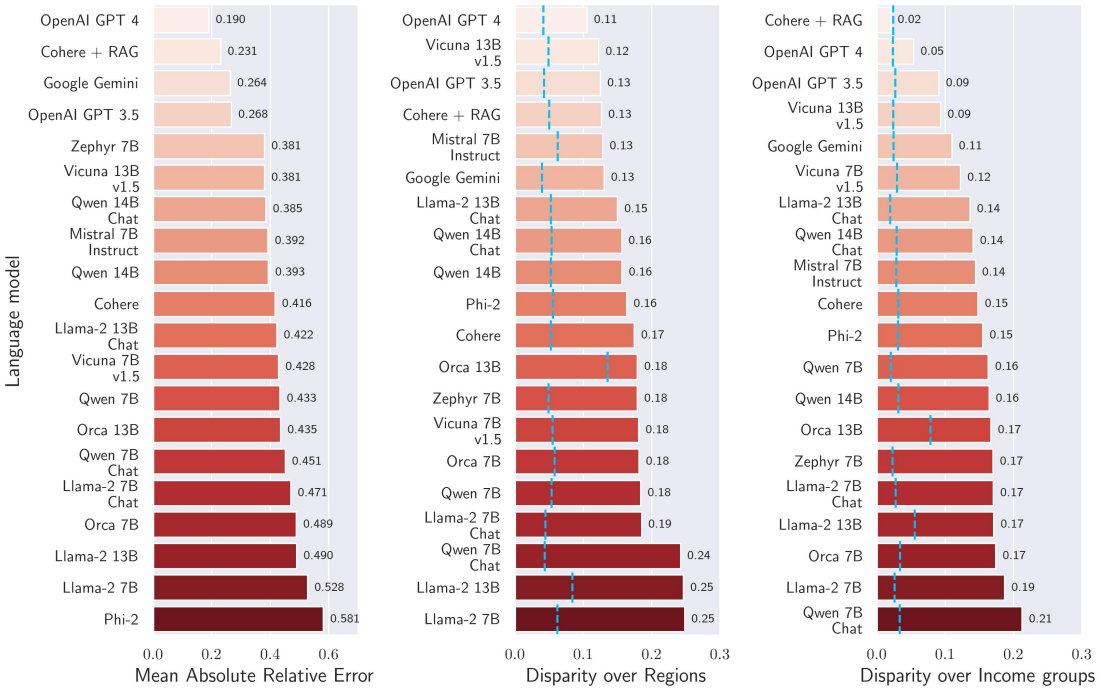


Figure 10.7: Performance of 20 LLMs averaged over 11 indicators from WORLDBENCH. We present the absolute relative error (left), as well as disparities across regions (middle) and income groups (right). For disparities, the blue dashed lines correspond to the disparity incurred using a random categorization of countries (into 7 groups for Regions and 4 for Income groups), averaged over ten trials. Observed disparities far exceed the amount expected for a random categorization of countries across nearly all LLMs.

Namely, we define *Disparity* as $\max_{e_i, e_j \in E} e_i - e_j$, where E is the set of mean absolute relative errors for each category of a given categorization. In other words, for example, *Disparity* over *Regions* is the gap between the mean absolute relative errors for the region with the greatest error and the region with the least error. Disparity also always falls between 0 and 1. To contextualize disparity scores, we compute a baseline corresponding to the disparity achieved using a random categorization of countries into k groups; we set $k = 7$ for Regions and $k = 4$ for Income groups. We approximate the baseline disparity given a set (i.e. for one LLM of interest) of per-country errors by applying ten random country categorizations and averaging the observed disparity over all trials.

Figure 10.7 visualizes average error and disparities per LLM. From the left most panel, we

see that the lowest mean absolute relative error achieved is 0.19, and the value for most models is near 0.4, indicating that there is substantial room for improvement for this task. Shifting from error to disparity (middle and right panels), we observe that **all models exhibit disparate performance over regions**, with gaps of at least 0.1 between the regions with the most and least error per LLM. Across income groups, disparities are also consistently present, though to a lesser degree, with only 4 of the 20 models studied achieving a disparity below 0.1. Nonetheless, both over Regions and Income groups, observed disparity almost always far exceeds the expected disparity for a random categorization (blue dashed lines).

A few expected trends emerge: base models are outperformed by their chat-tuned versions; smaller models are outperformed by their larger versions. One such trend we highlight is the impact of retrieval augmented generation (RAG), which is utilized for to augment the Cohere LLM. Incorporating RAG reduces mean absolute error by nearly a factor of two, reducing it from 0.416 to 0.231. Impressively, RAG causes disparity across Income groups to nearly vanish, going from 0.15 to 0.02, the lowest such disparity observed across our model suite, and *on par with a random categorization of countries*. However, it is worth noting that RAG comes at the cost of latency, as internet searches are required and the LLM must review retrieved documents in addition to the provided prompt. Nonetheless, RAG appears to be a promising direction for reducing errors and also potentially disparities.

Turning our attention now to indicators, Figure 10.8 shows errors and disparities per indicator. Mean absolute relative error exceeds 0.3 for all but two of the indicators. Again, disparities are present for most cases, though they are more pronounced across Regions than across Income groups. Moreover, over both Regions and Income groups, *observed per-Indicator disparity far exceeds the random baseline in almost all cases*. Indicators that seem to be driving the observed

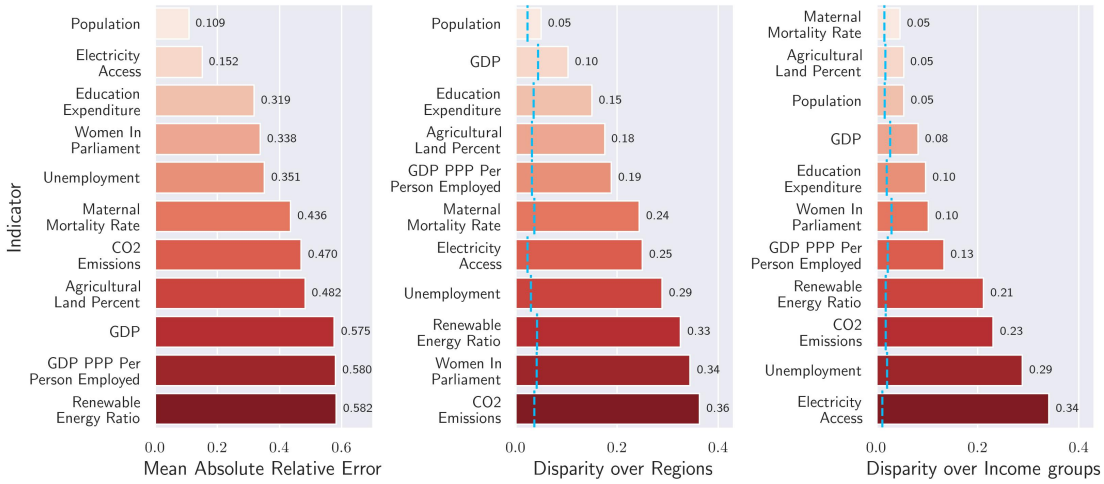


Figure 10.8: Error rates and disparities per indicator, averaged over LLMs. For disparities, the blue dashed lines correspond to the disparity incurred using a random categorization of countries (into 7 groups for Regions and 4 for Income groups), averaged over 10 trials.

disparities include CO_2 Emissions, Renewable Energy Ratio, and Unemployment.

10.5.4 Ordering of Regions and Income groups by error are consistent per-LLM and per-Indicator: Lowest error is with Western and high income groups

Having demonstrated that significant disparities are present for each LLM and each indicator separately, we now show that the order of disparity is consistent as well. that is, the regions and income groups with highest and lowest error (respectively) are the same within each subset. Namely, LLMs achieve the lowest error when answering questions about Western or high income countries, and they suffer the greatest error when answering questions about countries from the low income category. In figure 10.9, we show the distribution of *error ranks*. That is, e.g. in the top right heatmap, for each LLM, we rank the regions by their mean absolute relative erros, and then report the fraction of LLMs for which a region obtains a specific rank. Thus, we see that

for 75% of the LLMs, the highest error occurs for Sub-Saharan African countries. Strikingly, the pattern across income groups is strongly pronounced. Error ranks are almost perfectly inversely related to amount of income, wiht the high income group having lowest error for 95% of LLMs and the low income group having highest error for 90% of LLMs. Again, the same trends emerge when inspecting error rates per indicator.

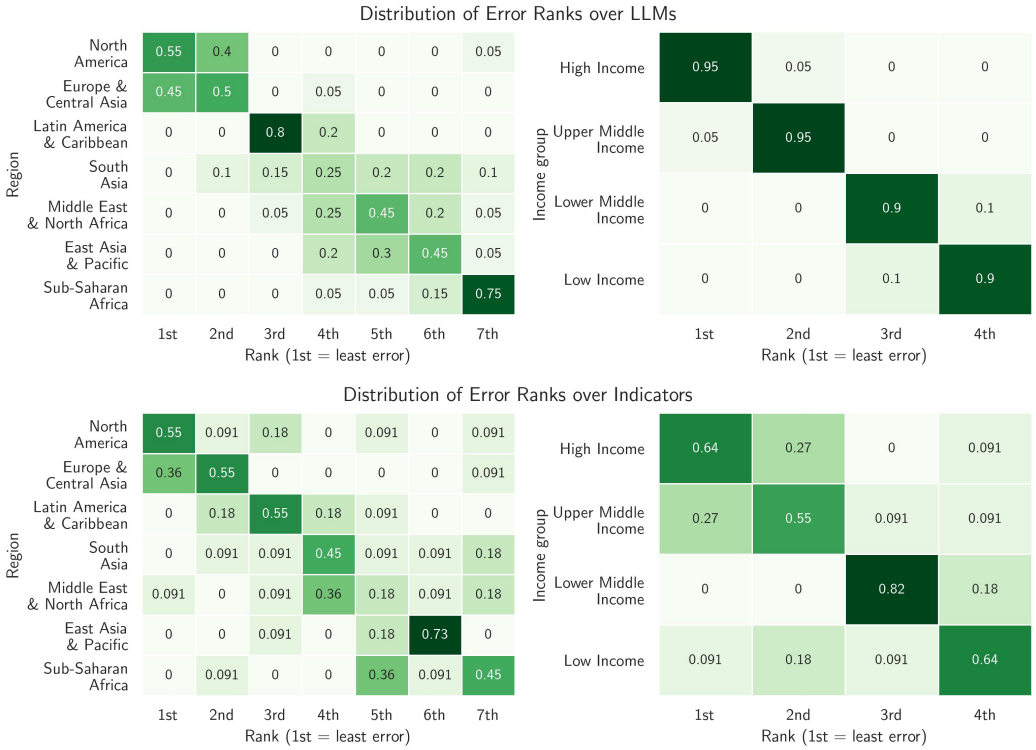


Figure 10.9: The order of regions and income groups by absolute relative error is largely consistent per LLM (top) and per indicator (bottom). For both LLMs and indicators, the regions with the lowest errors are most frequently North America and Europe & Central Asia, while the regions with the highest error are most frequently Sub-Saharan Africa and East Asia & Pacific. For Income groups, error nearly always increases as income decreases.

Thus, the original trends we observe when averaging over all LLMs and indicators, visualized in Figure 10.5, appear to hold when we inspect each LLM individually and each indicator individually. These results suggest that geographic and income-based disparities in LLM factul recall are pervasive throughout existing LLMs.

10.6 Noteworthy Observations

In analyzing per-country performance and geographic disparities in LLM factual recall, we additionally came across a number of noteworthy observations made possible by our benchmark. First, we found that LLMs occasionally offer what resembles citations in their responses, including instances where the WorldBank itself was mentioned. Since we have that exact data, we were able to cross-check the LLM "citations". Second, because we have data per-country *per-year*, we could compute error rates while selecting groundtruths from specific years, so to see how up-to-date LLM responses are. We explore these observations in more detail below.

10.6.1 Citation Hallucination

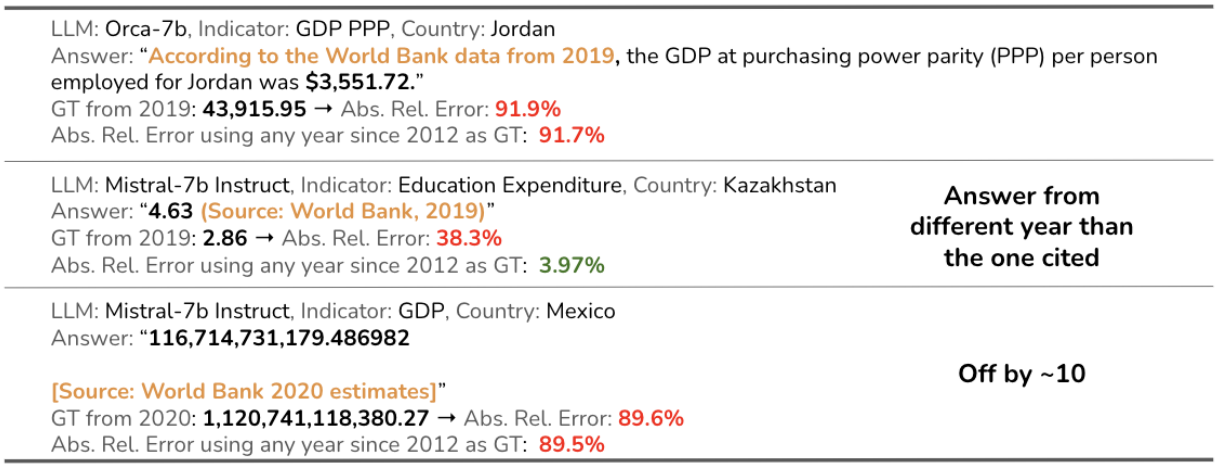


Figure 10.10: In addition to hallucinating false answers, we also observe LLMs to occassionally hallucinate *citations*. Above, a few examples of hallucination citation are shown.

Despite being prompted to only return a numeric value, the LLMs we studied still would often produce additional text. Interestingly, sometimes generated text would resemble a citation[1], claiming the provided answer was sourced from institutes like the World Health Organization, the

[1]Such behavior has been observed in [294]

International Monetary Fund, and even, the World Bank. In the last case, we cross-checked the provided responses to see if the numeric response matched the groundtruth World Bank data, contained in WORLDBENCH. Overall, *responses with "citations" were no more accurate than those without "citations"*, still incurring substantial mean absolute relative errors. Specifically, in 650 instances where the string "World Bank" (case insensitive) was mentioned, mean absolute error rate was 0.465. This suggests that the LLM-produced "citations" are hallucinated, as the provided responses do not actually come from the sources listed. Figure 10.10 displays a few examples of LLM produced "citations". For each example, we highlight the "citation", and provide the absolute relative error of the parsed answer compared to (1) the groundtruth value from the specific year cited, and (2) the lowest absolute relative error to groundtruths for any of the past ten years. In the first example, the LLM answer is way off, despite the arguably convincing "citation". Interestingly, we also observe an instance where the provided answer does not match the groundtruth from the cited year, yielding an error of 0.383, but it does match the groundtruth from the following year, with error dropping to 3.97%. Finally, we see an example where the provided answer is off by almost exactly a factor of 10 (relative error of ~ 0.9). This highlights a pitfall in using LLMs to return numeric information, as the difference in tokens between two numbers can be very small, while the resultant encoded value can be very large.

In summary, hallucinated citations pose a serious challenge in LLM reliability. On one hand, producing false citations obfuscates model errors, and denigrates the overall user trust in the system. On the other, that the LLMs appear to know what sources would contain the answer seem to be an encouraging sign to the potential benefits of retrieval-augmented systems.

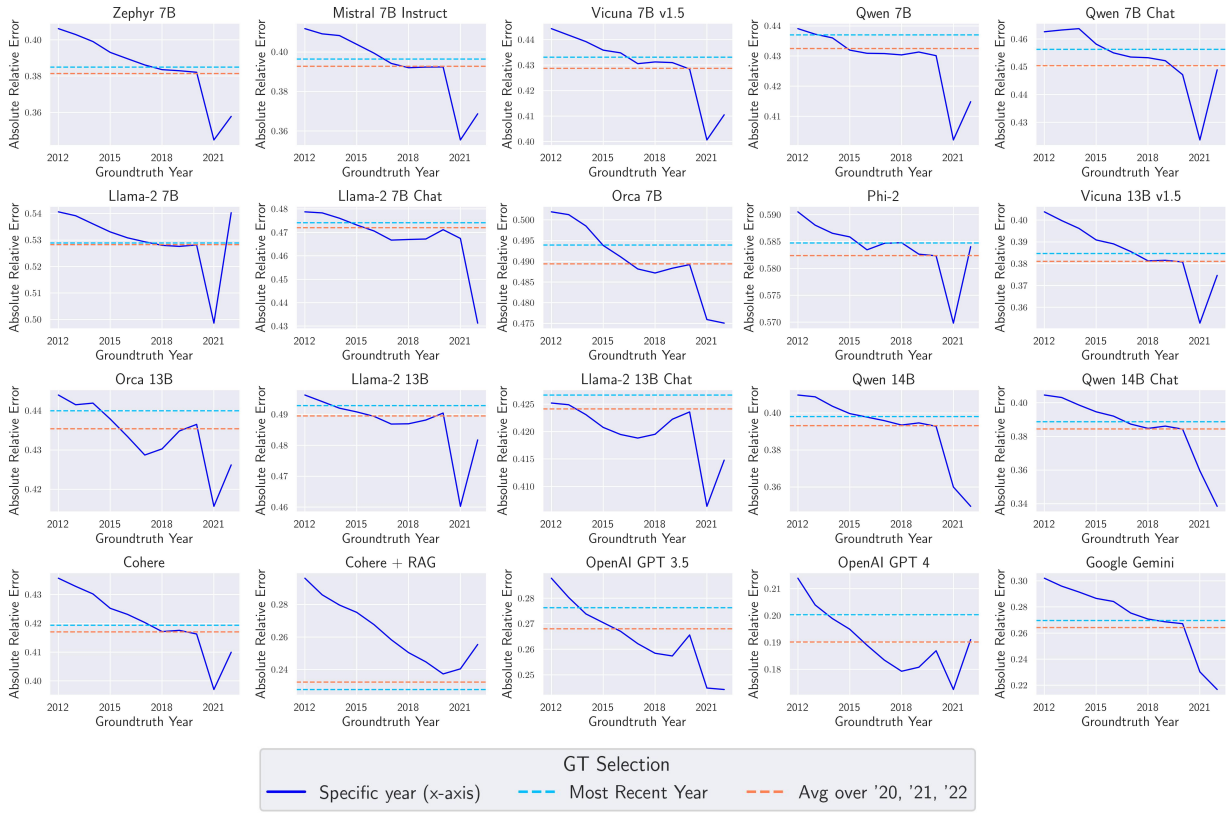


Figure 10.11: Error rate of LLM outputs compared to year from which groundtruth is extracted. Many models show the lowest error rate when their outputs are compared to groundtruths from 2021, indicating that models may already be slightly out of date.

10.6.2 Are some LLMs already out of date?

Now, we compare LLM responses to groundtruths from specific years for all LLM responses, not just the rare few where "citations" are present. Figure 10.11 shows the mean absolute relative error over indicators and all countries per LLM, computed using groundtruths selected in a variety of ways. The orange dashed line corresponds to the default groundtruth selection (averaging over any available data from the past three years), while the light blue one corresponds to using data from the most recent year (per country). The solid blue lines correspond to using the groundtruth value from the year on the x-axis. A trend that emerges in 13 of the 20 LLMs is that the lowest error occurs when comparing to data from 2021. In one extreme, error increases from

Correlation with Indicator			Self-consistency		
Indicator	Pearson's r	p-value	Indicator	Pearson's r	p-value
Maternal Mortality Rate	0.383	0.029	Population	0.749	0.000
Renewable Energy Ratio	0.135	0.180	Electricity Access	0.653	0.000
Unemployment	0.078	0.344	Women In Parliament	0.386	0.036
Education Expenditure	-0.047	0.453	CO2 Emissions	0.338	0.012
Agricultural Land Percent	-0.075	0.372	Renewable Energy Ratio	0.337	0.004
Population	-0.138	0.112	Education Expenditure	0.325	0.016
CO2 Emissions	-0.189	0.106	Agricultural Land Percent	0.324	0.057
GDP	-0.194	0.074	Unemployment	0.319	0.004
Women In Parliament	-0.344	0.046	GDP	0.246	0.136
Electricity Access	-0.348	0.005	Maternal Mortality Rate	0.155	0.245
GDP PPP Per Person Employed	-0.396	0.015	GDP PPP Per Person Employed	-0.016	0.206

Table 10.2: (**Left**) Correlation between per-country mean absolute relative error and individual indicator values. (**Right**) Per-indicator, correlation between per-country mean absolute relative error and normalized standard deviation of responses obtained over five trials.

0.5 to 0.54 when changing the groundtruth year from 2021 to 2022. These results suggest that the facts internally stored in some LLMs may already be out of date, reporting statistics closer to previous years, especially if their training data was curated in years past. Of course, an LLM cannot recall a fact that did not exist at the time of its training. Nonetheless, as the use of LLMs continues to grow, the ability to stay up to date will be paramount. We hope WORLDBENCH can aide in this pursuit.

10.6.3 What kinds of countries experience high error rates?

We now present a purely correlational study to better understand what countries experience the highest error rates. Using the per-country data for each indicator studied, we compute the correlation between these values and per-country error. We also compare the normalized (by mean) standard deviation of responses per country per indicator, with responses taken over five trials. The hypothesis here is that LLMs will have greater variance in answering questions about countries they are less accurate for, similar to [294]; we call this self-consistency. We compute

correlation to country-wise errors for each (LLM, indicator) pair separately, as the values can take on substantially different ranges as either LLM or indicator changes, and then average over all such pairs. Results are reported in table 10.2. We find that most indicators are not correlated with per-country error. The strongest correlation is -0.396 for GDP PPP per person employed, suggesting that LLMs perform worse on countries with lower per-person wealth. Notably, neither population nor GDP are correlated well with error. As for self-consistency, in most cases, correlations are within $0.3-0.4$. In a couple instances, high correlations are observed, suggesting that sampling multiple outputs and inspecting variance can sometimes (but not always reliably) aide in estimating the uncertainty of the LLM.

In summary, our simple correlational analyses do not shed much insight in to why particular countries incur higher error rates for LLMs. We conjecture that the availability of training data plays a large role. However, the groundtruths are available *for all countries*, and World Bank data is likely in the training sets of many LLMs, as indicated by the hallucinated citations to them. We leave investigation to the cause of the geographic disparities we observe to future work.

10.7 Limitations

Is it reasonable to expect language models to perform this task? LLMs are not directly optimized for information retrieval, and developers often caution that LLMs many not always provide factual answers. Furthermore, retrieving specific numbers can be challenging, given the fact that many sequences of numbers are feasible/reasonably likely to appear in natural language, where as the distribution of words has far less entropy. Nonetheless, LLMs have been observed to produce factual responses to certain queries, achieving as high as 86% exact match on TriviaQA

[295]. Indeed, in our experiments, we observe mean absolute error rates as low as 3.6% for the Population indicator and 5.8% for the Electricity Access indicator, suggesting that LLM-based factual recall is feasible. We emphasize that the point of our benchmark is to enable comparison in LLM performance across countries, so to uncover systemic disparities. Moreover, despite warnings from developers, as LLMs become more ubiquitous, end users will likely still make factual queries, to which we'd hope language models respond accurately, and importantly, without substantial differences in performance due to factors like geography or wealth of the country of interest. We hope our benchmark will help to ensure that LLMs exhibit fair performance.

Can LLMs ever ace this task? Some of the indicators studied are volatile, in the sense that they change non-trivially from year to year. Also, some metrics can take on slightly different values based on which organization measured them (e.g. the World Bank's numbers may differ from the United Nation's numbers). Thus, we do not expect LLMs to achieve perfect performance on this metric. Nonetheless, we believe our benchmark can offer valuable signal in measuring geographic disparities. That is, even though error rates may never be exactly zero, we can hope that they will not vary substantially across countries.

10.8 Conclusion

We present WORLDBENCH, a benchmark to quantify geographic disparities in LLM factual recall. We find pervasive and consistent biases across 20 evaluated LLMs, with Western and higher income countries experiencing lower error rates. By utilizing World Bank data, our benchmark is flexible and will remain up to date. Thus, we hope our benchmark can aide in reducing geographic disparities of future generations of LLMs, towards models that work well *for all*.

Chapter 11: Embracing Diversity: Interpretable Zero-shot classification beyond one vector per class

Now, we investigate an analogous geographic disparity to the one we saw in the previous chapter. Namely, we study geographic disparities for the familiar task of image classification. However, we focus on the case where vision-language models are employed to perform classification in a zero-shot manner. The ability of these models to accept open-vocabulary language inputs makes them a fantastic candidate for collaboration with an auxiliary LLM. We put these components together to understand and mitigate the geographic disparities we observe, resulting in a more transparent, fair, and performant zero-shot classifier. The following is drawn from a FAccT '24 paper co-authored with Michael Rabbat, Mark Ibrahim, and Diane Bouchacourt [200].

Vision-language models enable open-world classification of objects without the need for any retraining. While this zero-shot paradigm marks a significant advance, even today's best models exhibit skewed performance when objects are dissimilar from their typical depiction. Real world objects such as pears appear in a variety of forms — from diced to whole, on a table or in a bowl — yet standard VLM classifiers map all instances of a class to a *single vector based on the class label*. We argue that to represent this rich diversity within a class, zero-shot classification should move beyond a single vector. We propose a method to encode and account for diversity within a class using inferred attributes, still in the zero-shot setting without retraining.

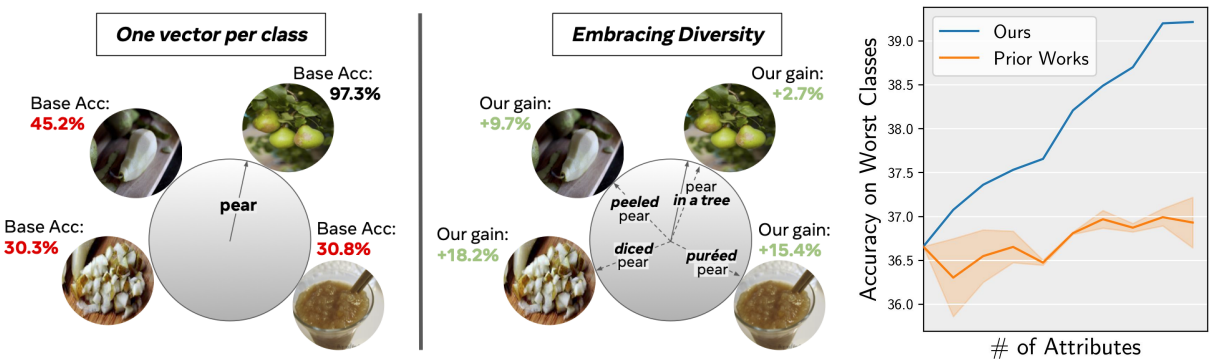


Figure 11.1: (**Left**) Instances of a class can appear in many diverse ways, like the pears above. Using one vector (the classname embedding) to represent the whole class results in disparate performance, particularly for atypical instances. (**Middle**) To address this issue, we infer attributes and embed multiple vectors, reducing disparities and enhancing interpretability. (**Right**) Our method scales better than prior works as we include more attributes (Section 11.6.1), enabling us to account for the many ways in which diversity can arise.

We find our method consistently outperforms standard zero-shot classification over a large suite of datasets encompassing hierarchies, diverse object states, and real-world geographic diversity, as well finer-grained datasets where intra-class diversity may be less prevalent. Importantly, our method is inherently interpretable, offering faithful explanations for each inference to facilitate model debugging and enhance transparency. We also find our method scales efficiently to a large number of attributes to account for diversity—leading to more accurate predictions for atypical instances. Finally, we characterize a principled trade-off between overall and worst class accuracy, which can be tuned via a hyperparameter of our method. We hope this work spurs further research into the promise of zero-shot classification beyond a single class vector for capturing diversity in the world, and building transparent AI systems without compromising performance.



Figure 11.2: We test models on datasets that provide groundtruth **attributes** (shown in bold) annotating hierarchies, diverse states, and real-world shifts (e.g., Rojas et al. [2] labels the income level and country of origin of each image, towards promoting AI models that reduce bias) within a class. We find that standard zero-shot accuracy ('Base Acc.' above) drops significantly when certain attributes are present, namely when the attribute manifests in visual differences from what the model considers 'typical' for the class. We design our method to improve performance on these 'atypical' instances.

11.1 Introduction

A pivotal advance in machine learning is the advent of *foundation models*. A single foundation model trained on large-scale data can supplant multiple task-specific models. Vision-Language models (VLMs) are popular foundation models capable of encoding text and images in the same representation space. Compared to standard classifiers which can only classify objects from a predefined list of classes with examples, VLMs are capable of open-world, zero-shot classification—meaning, VLMs can classify any object using text descriptions without any additional training. This zero-shot paradigm has spurred the development of many VLMs [13, 296, 297] with impressive classification performance.

Despite their remarkable performance, even today's best models exhibit skewed performance for certain groups of images. For example, [298] show models such as CLIP have exacerbated the gap in performance between regions such as Africa and Europe (as well as the gap across income-levels). We find similar biases arise when an object is visually dissimilar from its

typical depiction. For example, Figure 11.1 (left) shows CLIP's 97.3% accuracy on typical pears drops dramatically when a pear is peeled (45.2%) or puréed (30.3%). Addressing such biases is crucial to the reliability of classifiers in the real world, where instances within a class can vary significantly.

Zero-shot classifiers, like standard models, use a single vector in deep embedding space to describe an entire class. For standard zero-shot classification, a vision-language model (i) encodes the image along with 80 hand-crafted prompts per class name (e.g., "a photo of a pear" or "a drawing of a pear"), (ii) averages the 80 embeddings per class to obtain a single vector, (iii) predicts the class whose vector maximizes cosine similarity to the image embedding [13]. Prompt averaging encourages all instances of a class to be mapped to the same vector in the model's embedding, inherently limiting the model's ability to infer the innumerable diversity within a class. A pear can be diced, sliced, whole, in one's hand, or in a bowl. In each case, the image of the pear would be markedly different, and its embedding may not always be well aligned with the single vector that is supposed to represent the *entire* class. Thus, there is a natural tension between the one vector per class paradigm and performing consistently across a class with high diversity, which we empirically validate.

While many strategies exist to mitigate performance disparities when labeled-data is available, these methods do not transfer to the data-free setting of zero-shot classification. Fortunately, unlike standard classifiers, the open-world nature of VLMs enables them to represent any attribute using the text encoder. VLMs can enrich the single per-class vector with attributes to more faithfully capture the variety with which a class can appear, pinpointing whether a pear is peeled or puréed. Thus, we argue that instead of learning one vector per class that is invariant to diversity, we should leverage the open-world nature of VLMs to *explicitly account for* the diversity within

a class (i.e., via multiple vectors).

Recent works offer promising signs that zero-shot classification can be improved by incorporating attributes beyond the class name, such as subclasses [299] or visual descriptors [300, 301]. However, the former is limited to datasets with hierarchical label sets, and the latter reverts back to the one vector per class paradigm via simple averaging, limiting the benefits of incorporating more attributes (Section 11.6.1). Importantly, diversity comes in many forms that generic descriptors or subclasses alone may not adequately capture.

In this work, we propose a zero-shot method for enriching classes with open-ended attributes to boost zero-shot classification. Our method consists of two steps: (i) an attribute inference step, in which we use generative language modeling (an inherent, under-utilized capability of some modern VLMs) to enumerate relevant attributes along *many* axes of diversity, and (ii) a prediction consolidation step, where we flexibly attend only to subpopulations (i.e., instances within a class sharing an attribute) that are most relevant to the image. By enriching and carefully consolidating attributes to describe diversity within a class, our method more faithfully encodes atypical instances. Furthermore, by introducing interpretable intermediate outputs (i.e. the inferred attributes), our method affords greater transparency, as each inference comes with the specific list of fine-grained attributes used to predict the class, and attribute overlaps across classes can help anticipate and articular potential failures, before they happen.

In experiments over a large suite of datasets encompassing hierarchies, diverse object states, and real-world geographic diversity, we observe our method matches and in most cases exceeds the performance of existing methods, showing that transparency can be achieved without compromising on performance. Our method yields consistent gains on a second dataset suite with finer-grained classes and no labeled diversity, showing that our method still works well when

intra-class diversity may be less present. Encouragingly, we find larger improvements occurring for the hardest classes and subpopulations, where atypical instances are usually found, resulting in reduced performance disparities. Compared to existing methods, we find that our approach can effectively scale to a much larger number of attributes to cover broader axes of diversity, as shown in the right panel of Figure 11.1. Our method also offers a principled trade-off between accuracy overall vs. on the worst classes, all without additional training. In summary, we (i) identify a limitation of the one-vector-per-class paradigm in adequately representing classes with diverse subpopulations, (ii) propose to go beyond one vector per class, leveraging under-utilized abilities of VLMs to explicitly account for intra-class diversity, and (iii) extensively validate the effectiveness of our method to perform zero-shot classification in *both* a more transparent and accurate way, especially for diverse subpopulations that are often overlooked.

11.2 Review of Literature

Despite impressive overall accuracy, modern classifiers still suffer from biases. That is, they under-perform on some parts of the data, often due to spurious correlations or data imbalances in the training set. These biases can result in significant negative real-world impact. For example, Buolamwini and Gebru [9] exposed significant bias along demographic lines for facial recognition systems, and more recently, Richards et al. [298] demonstrated that despite steady progress on typical benchmarks, today's best models still generalize poorly to images from lower-income households and certain geographic regions. Namely, VLM-based zero-shot classifiers were shown to have even larger performance disparities across geographic and economic shifts than their supervised counterparts.

However, the promise of open-world zero-shot classification rightfully draws much attention to VLMs, which operate by mapping images and text to a shared latent space. CLIP [13], a seminal VLM, achieves this via joint contrastive training of image and text encoders on 400 million image-caption pairs. Recent models such as BLIP-2 [296] bootstrap the training of more powerful VLMs by taking larger pretrained vision and language backbones and fusing their outputs to a single space, which in turn can even be used to generate text; that is, some modern VLMs contain a fully functional LLM with (often under-utilized) generative abilities. To perform zero-shot classification with VLMs, one computes the class that has the highest cosine similarity between a test image's embedding and the embedding of a class name, often averaged over many (80 for CLIP) handcrafted prompt templates. While many efforts have improved VLM-based classification via prompt-tuning [302, 303, 304, 305, 306, 307, 308], nearly all require some labeled data. Other works focus more closely on the task of debiasing VLM-based classifiers [246, 309, 310, 311], though they too utilize labeled data, placing them out-of-scope of the true zero-shot setting.

Compared to previous classifiers, the key novelty of VLMs is their ability to encode *any* text. However, standard zero-shot classifiers only embed classnames, either alone or averaged over prompts. We propose to leverage the open-vocabulary capabilities of VLMs to improve coverage of intra-class diversity by embedding more than just the class name. One effort along these lines is PerceptionCLIP [312], which infers contextual attributes per image as generative factors and does class inference conditioned on them. Other works utilize LLM-generated class descriptors, towards creating a concept-bottleneck [313] or rationales for inference [314], though these methods use data to train a linear layer atop descriptor similarities. DCLIP [300] show including descriptors can also improve performance in the zero-shot setting, and Pratt et al. [301]

extend the gains using additional handcrafted queries. WaffleCLIP [315] shows that appending random characters or words can achieve similar performance to descriptor-based methods like DCLIP, without the need for an external language model. Importantly, although these works obtain more than one vector per class, they ultimately average over them. Thus, decision boundaries remain linear and biases may linger, as atypical instances are still suboptimally uncovered (see Sections 11.4.2 and 11.6.2). In contrast, like us, CHiLS [299] introduces a *non-linearity* in three steps: they (i) define subclasses with groundtruth label hierarchies or by querying GPT-3, (ii) do zero-shot classification on this extended set of classes (subclasses) and original classes, (iii) reweight the standard zero-shot score for each class with the max score from step (ii) over subclasses within the class. However, CHiLS is designed specifically for hierarchical label sets, which limits the types of diversity it can capture (see Section 11.6.1).

11.3 Motivation

We hypothesize that the standard one-vector-per-class paradigm poses a tension for highly diverse classes. We investigate this by measuring classification performance as a function of class diversity. Indeed, we find classes with higher diversity suffer worse performance under the one-vector-per-class classification paradigm. Then we illustrate how newfound open-vocabulary capabilities of VLMs can enrich the single class vector to encompass diverse instances without additional training. That is, we show that incorporating attribute information can substantially improve VLM recognition of atypical subpopulations.

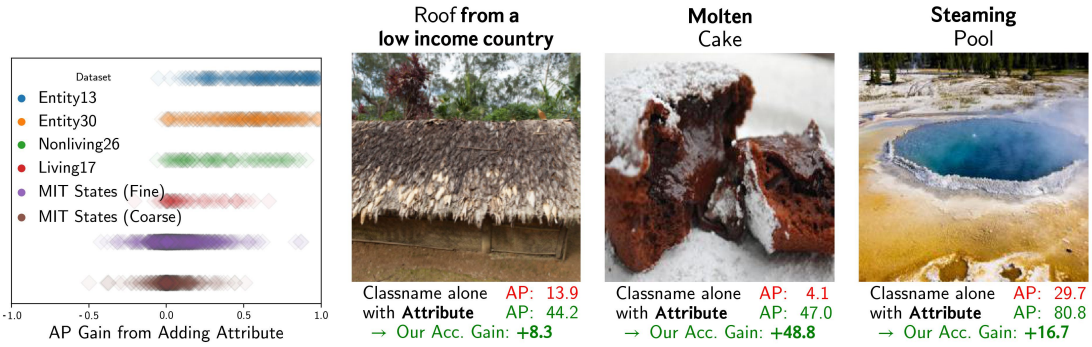


Figure 11.3: The average precision (AP) of a classname embedding is often much lower than the average precision of a subpopulation (i.e. classname with attribute) embedding. Subpopulations that see large increases in AP by including the attribute tend to be atypical. We design our method to improve accuracy on these diverse subpopulations, by inferring and explicitly accounting for them.

11.3.1 A single vector inadequately represents diverse classes

A standard VLM classifier is most effective when it aligns all instances of a class to their class vector (and away from vectors for other classes). Intuitively, aligning instances with high diversity is challenging as their image embeddings are more dispersed—and particularly tough for fixed open-vocabulary VLMs that do not benefit from knowing the specific classes of interest during their pre-training. We see in Figure 11.2 for example the less typical *Arctic* `fox` is far harder to recognize than a typical `fox` (52.0% versus 84.5% accuracy). We observed similar drops in accuracy for a *deflated* `balloon` versus a regular `balloon` and an *unpaved* street versus a paved one. To systematically quantify this tension, both for VLMs and for the one vector per class paradigm generally, we examine class accuracies on ImageNet [14] relative to the diversity of each class across several models with varying levels of supervision. To proxy diversity, we measure the variance of image embeddings within a class. In all cases, we observe a strong negative correlation between class-wise accuracy and diversity. That is, **classes with higher diversity have lower accuracy in the one vector per class paradigm**.

11.3.2 A path forward: VLMs can recognize diversity with relevant attributes

Although standard VLMs use solely classname in zero-shot classification, their shared embedding space allows to encode relations to any other text. In turn we ask: *can the open-vocabulary encoder of VLMs better situate diverse classes given relevant attributes?* Specifically, we assess whether enriching classes with attributes can improve zero-shot classification on a suite of datasets with ground-truth attributes per class. We form a **subpopulation** by taking instances within a class that share an attribute. For each subpopulation, we compute the similarity of image embeddings with the text embedding of (i) the classname and (ii) the classname with the corresponding attribute, using CLIP ViT-B/16. We then measure the average precision of the two similarity scores for distinguishing instances within the subpopulation from instances outside of the class. We find, as shown in Figure 11.3, that for the vast majority of cases, incorporating attributes leads to more precise recognition, and often by large margins: adding `molten` to `cake` improves average precision by over 40 points. Upon inspection, the highest gains in average precision tend to occur for atypical subpopulations. Thus, VLMs *can* recognize instances in a class even when they are atypical, but this ability is restricted under the one vector per class paradigm.

11.4 Method

We now propose a method to better utilize the ability of VLMs to recognize diverse subpopulations. Our method consists of **attribute inference** and **prediction consolidation**. First, we query a large language model (LLM) for diverse per-class attributes that span many (often overlapping) subpopulations. Then, after computing the similarity of an image to each subpopulation, we non-linearly consolidate these similarities to obtain one score per class. We elaborate

on these two steps below.

11.4.1 Attribute Inference Along Many Axes of Diversity

To better cover the diverse subpopulations that may exist within a class, we incorporate attribute information. However, **diversity can come in many forms**. That is, the way in which two instances of a class differ can itself vary. Consider the examples in Figure 11.2. The *Arctic* `fox` case shows how a class can contain distinct finer-grained categories. In a related manner, the state or condition in which the class instance is in can also substantially change its appearance: a `balloon` looks much different when it is *deflated*. Further, there exist generic attributes that can lead to substantial visual differences regardless of the class, such as the region or income level of the country where an image is taken, exemplified by the two `Street View` images. Thus, to capture the many ways in which diversity can arise, we employ multiple distinct queries, in contrast to prior work. Namely, we infer:

- *Class specific* attributes, such as the possible **states** of an object (e.g., *diced* or *sliced* for `pear`). We also obtain **descriptions** for and different **kinds** of each class, as in DCLIP and CHiLS respectively.
- *Class adjacent* attributes, like **co-occurring objects** or **backgrounds**, to get useful context.
- *Class agnostic* attributes that describe how objects vary in general. For example, towards improving geographic fairness, we list potential choices for the **income-level, region** and **country** of origin of the image. We also introduce a novel two-step LLM query, where we first ask the LLM to list generic axes of diversity, and then have it populate those axes. We name this **auto-global** as it automatically generates many global attributes.

11.4.2 Nonlinear Prediction Consolidation

Enumerating attributes along various axes of variation results in descriptions of many diverse subpopulations per class. Since VLMs have open-vocabulary text encoders, we can directly embed these subpopulation descriptions, in addition to the class name. Given a test image, we compute similarities to each of these embeddings. We then must consolidate them to obtain a single score per class.

Figure 11.4 illustrates the simple case of `fox` vs `wolf` classification, where solid/dotted lines correspond to classname/subpopulation embeddings on the hypersphere (shown here in 2D). The left-most panel shows examples from the two classes near where their image embeddings would lie. Text embeddings for the subpopulations (dotted lines) are close to corresponding image embeddings, as VLMs are capable of recognizing even diverse subpopulations (see Section 11.3.2). Standard zero-shot inference maps a test-time image to the class of the nearest classname text embedding. Since there is only one vector per class (the classname-based embedding), the decision boundary is linear, as shown in the middle panel. The edge of the hypersphere is colored (orange for wolf, blue for fox) to indicate the predicted class for an image embedding at that location. Notably, the *Arctic* `fox` is misclassified as `wolf`, as its appearance more closely resembles a typical wolf than a typical `fox`, and so, the embeddings of *Arctic* `fox` images fall closer to the text embedding of "wolf" (and vice-versa for the *red* `wolf`). Methods like DCLIP and WaffleCLIP embed more than just the classname, but they consolidate similarities via averaging, again resulting in a linear decision boundary. Even if atypical subpopulations are included at first, averaging can narrow the initial diverse coverage, as most embeddings for a class may better describe a typical instance.

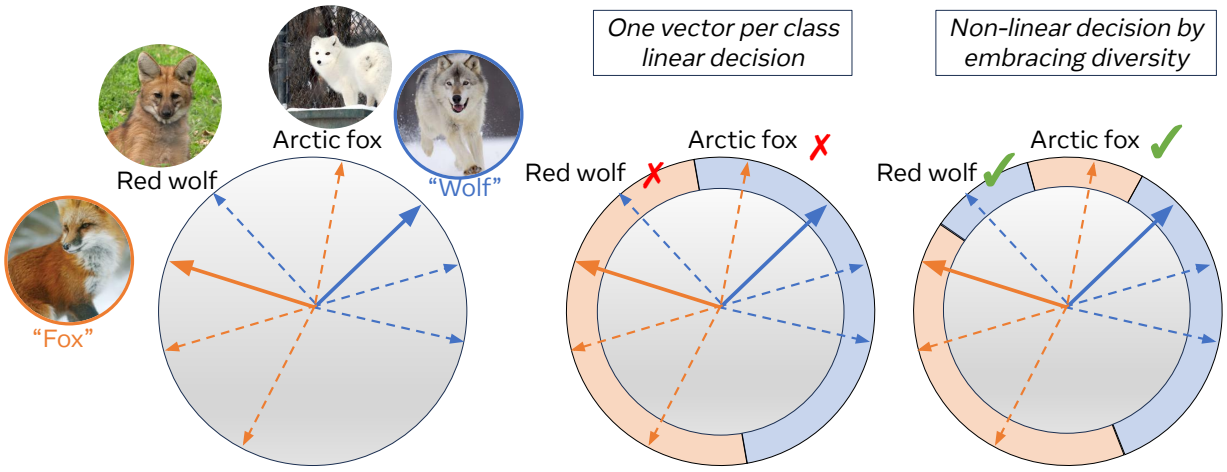


Figure 11.4: An *Arctic* `fox` can more closely resemble a typical `wolf` than a typical `fox`. Standard zero-shot classification using one vector per class (the classname embedding) is ill suited for this case. We address this issue by nonlinearly consolidating similarities to *multiple* vectors per class that explicitly encode the diverse subpopulations within the class. See section 11.4.2 for full explanation.

In contrast, we propose the following *nonlinear* consolidation: we compute the single score per class for a given test image as the average of the similarities of the image embedding to *only the* k *closest* subpopulations embeddings for the class, where k is typically small (we use $k = 16$). This way, an image can have a high class score even if it is only similar to a small subset of subpopulations, as is the case for atypical instances. Thus, the *Arctic* `fox` and *red* `wolf` can be correctly classified despite being far from the classname and most subpopulation embeddings for their respective classes, as shown on the right panel of Figure 11.4, where we use $k = 1$ for simplicity (i.e. images are mapped to the class of the closest dotted or solid line, leading to a non-linear boundary). We shed insight on the effect of varying the hyperparameter k in Section 11.6.2, revealing a tunable accuracy-fairness trade-off.

11.5 Analysis

We now empirically validate our method's effectiveness and enhanced interpretability over two dataset suites. Our method performs on par with (and usually surpasses) existing approaches in overall accuracy. Notably, we see larger gains for the hardest classes and subpopulations, which are likely more diverse and atypical respectively (precisely the samples on which our method is intended to improve performance). Furthermore, while matching or exceeds the performance of existing, we offer unique interpretability benefits, such as fine-grained and faithful explanations, as well as the potential for *error anticipation*; we detail these below.

11.5.1 Consistent Gains Across Diverse Datasets, Particularly for the Hardest Classes and Subpopulations

11.5.1.1 Baselines and Metrics

We measure performance of zero-shot classifiers using the popular CLIP ViT-B/16 and BLIP-2 VLMs [13, 296]. To infer attributes, we utilize the open source Vicuna-13b-v1.5 language model [316], which notably is already contained in the BLIP-2 model we use. We report accuracy overall as well as averaged over the worst 20% of classes and subpopulations. Note that we only use groundtruth attributes when computing metrics; our method exclusively uses attributes *inferred* via the queries listed in Section 11.4.1. We also compute the lowest subpopulation accuracy per class and average that, so to obtain the metric denoted as 'Avg Worst Subpop'. For the real-world shifts, we also report worst region and worst income group accuracy. Our baselines include: standard zero-shot (only one vector per class, corresponding to the

classname embedding) which we call Vanilla, DCLIP (averages over class descriptors), Waffle-CLIP (averages over *random* descriptors sampled over ten trials), and CHiLS (reweights standard zero-shot class score with *max* probability of different kinds of the class). Notably, we average all text embeddings over the 80 prompts crafted for CLIP, so to report best possible baseline results.

11.5.1.2 Datasets

We curate a suite of eight *attributed* (so to have groundtruth subpopulations) datasets spanning different axes of diversity. We use the four Breeds datasets [172] for their hierarchical label sets, as used in the CHiLS paper; in fact, the Breeds datasets were the ones where CHiLS was most effective. Next, we devise two classification tasks (coarse and fine grained) from the MIT States dataset [317] to track performance over labeled states (e.g., *sliced* or *diced* for `pear`). Importantly, we also include the datasets Dollarstreet [2] and GeoDE [318], which contain images from varied geographic regions and income levels. As the diversity in these datasets is occurs naturally, they can encompass *many* axes of variation, as opposed to our other datasets that only varying along known axes, like object state or kind.

We also incorporate a second suite of the following 9 datasets *without* attributes: ImageNet [14], ImageNet variants (v2, -R, -A, -Sketch) [36, 84, 114, 319], Food-101 [320], Flowers-102 [321], FGVC-Aircraft [322], and Oxford Pets [323]. These datasets are all somewhat fine-grained[1], and as such, are less likely to have intra-class diversity than our original datasets. Thus, this dataset suite provides insight on if embracing diversity is only effective when diversity is to be expected.

[1]ImageNet (and its variants) contains 120 out of 1000 categories dedicated only to various dog breeds; Flowers-102 has many highly similar classes; Oxford Pets only has different breeds of dogs and cats; the FG in FGVC-Aircraft stands for fine-grained.

Dataset Type		Accuracy	Avg Worst Subpop	Worst 20% of Classes	Worst 20% of Subpops
States	Vanilla	66.71	40.66	35.46	21.73
	DCLIP	63.65	39.41	34.26	20.98
	Waffle	66.68	40.71	35.49	22.05
	CHiLS	66.56	40.41	36.16	22.45
	Ours	**67.92**	**41.53**	**38.16**	**23.64**
Hierarchical	Vanilla	78.15	48.36	50.72	35.89
	DCLIP	77.80	48.48	51.05	34.36
	Waffle	78.52	49.42	49.78	35.22
	CHiLS	79.44	**52.65**	51.80	38.44
	Ours	**79.50**	51.23	**52.59**	**38.57**

Table 11.1: Zero-shot classification on datasets with known variation types for CLIP with a ViT-B/16 encoder. States averages results over the two categorizations of MIT States data, while Hierarchical averages results over four Breeds datasets. We observe similar results for BLIP-2.

We note that we strived to minimally fit our method to the evaluation suite. That is, we do not optimize our query set to maximize performance on the datasets we selected, which can be challenging for zero-shot classification methods. One specific measure we took toward this end was fixing our method completely before evaluating on the second dataset suite. Thus, the second dataset suite serves as a held-out challenge set, intended to test the generalizability our method to settings where intra-class diversity may not be present.

11.5.1.3 Results

Table 11.1 shows results for datasets with diversity along hierarchical and states axes, and table 11.2 shows results for geographic diversity. Our method consistently matches (and even improves) accuracy of existing methods, even over CHiLS in the hierarchical setting it was specifically designed for. Notably, CHiLS becomes less effective for other datasets, while our method remains strong. We observe larger gains for worst class and subpopulation metrics, especially

DollarStreet Method	Accuracy	Worst Region	Worst Income	Avg Worst Subpop	Worst 20% of Classes	Worst 20% of Subpops
Vanilla	51.51	42.43	34.76	37.60	18.33	11.01
DCLIP	49.78	41.08	32.91	36.37	19.07	11.19
Waffle	51.37	42.71	34.97	37.69	18.12	10.74
CHiLS	51.68	42.20	33.90	37.60	20.51	12.72
Ours	**52.70**	**44.04**	**37.21**	**40.31**	**20.88**	**15.05**
GeoDE						
Vanilla	90.15	86.63	-	82.57	72.24	69.95
DCLIP	91.31	88.14	-	84.21	74.44	71.90
Waffle	91.59	89.06	-	**85.44**	75.85	74.37
CHiLS	90.96	87.90	-	84.48	73.27	71.64
Ours	**91.75**	**89.12**	-	85.40	**76.13**	**74.64**

Table 11.2: Zero-shot classification performance on geographically diverse images from DollarStreet and GeoDE using CLIP with a ViT-B/16 encoder. We see similar results for BLIP-2.

over baselines that consolidate via averaging (Vanilla, DCLIP, Waffle), supporting the claim that our method improves coverage of the most atypical instances, and that moving beyond the one vector per class paradigm helps in this regard. For example, compared to baselines that consolidate via averaging to obtain one vector per class, our method improves accuracy for the worst classes and subpopulations by $2-3\%$ in most cases. For Dollarstreet, these gains manifest in a 9% average relative gain over baselines for the accuracy over worst income group metric (and an even larger gain for the worst 20% of subpopulations), showing that our methodology can facilitate progress on real-world fairness indicators.

Turning our attention to the held-out challenge datasets, Table 11.3 shows our method can generalize effectively to finer-grained classification tasks where intra-class diversity is not explicitly known to be present. Our method improves accuracy on the hardest classes by an average of 1.5% over the closest baseline. Similarly, our method exceeds all baselines by about 1% in overall accuracy in nearly all cases, suggesting that embracing diversity does not come at a cost of

	ImageNet	v2	-A	-R	Sketch	Food	Flowers	Aircraft	Pets	Average
					Overall Accuracy					
Vanilla	68.48	61.98	30.16	59.24	48.37	88.35	66.09	31.26	**92.72**	60.74
DClip	68.85	62.37	31.35	60.04	48.54	88.05	**70.69**	32.67	92.23	61.64
Waffle	68.44	62.15	31.09	61.17	48.58	88.09	66.87	31.05	92.03	61.05
CHiLS	0.11	0.10	0.00	0.00	0.11	88.59	67.49	34.20	92.12	31.41
Ours	**69.94**	**63.32**	**32.19**	**61.49**	**49.38**	**89.06**	**70.69**	**34.62**	92.26	**62.55**
					Accuracy on Worst 20% of Classes					
Vanilla	34.42	25.75	7.47	29.27	9.19	73.58	2.08	0.00	**78.84**	28.96
DClip	34.58	25.65	6.24	28.90	8.99	72.92	3.44	0.00	77.50	28.69
Waffle	32.62	23.00	5.79	28.18	8.93	73.36	2.14	0.00	75.65	27.74
CHiLS	0.00	0.00	0.00	0.00	0.00	75.36	2.09	0.00	76.92	17.15
Ours	**37.30**	**27.60**	**7.98**	**33.74**	**9.33**	**76.18**	**4.40**	**0.25**	77.40	**30.46**

Table 11.3: Zero-shot classification performance on finer-grained **held-out** datasets without attributes, using CLIP with a ViT-B/16 encoder. We observe similar results for BLIP-2. Our method effectively generalizes to new settings without tuning the set of queries for attribute inference.

overall performance. Moreover, the effectiveness of our method in these new settings show that the queries we select (for inferring attributes) generalize beyond our original dataset suite. That is, we do not need to tune the LLM queries for each new classification task of interest. Nonetheless, the ability to add and remove LLM queries can be seen as a strength, as a practitioner is provided more control than in standard zero-shot classification.

11.5.2 Enhanced Interpretability at No Cost to Performance

11.5.2.1 Faithful Fine-grained Interpretations For Free

Having shown that our method is equally (and usually more) performant than existing approaches, we now discuss the enhanced interpretability of our method. Namely, each inference comes with a list of the k subpopulations specifically relevant to the test image for free. Figure 11.5 shows a few example where our method corrects misclassifications from the standard ap-

proach. These interpretations are faithful, as they are exactly the subpopulations used to compute the class score. Also, since we include attributes along various axes of diversity, our interpretations are finer-grained than prior work: DCLIP yields the same set general descriptors for any image predicted to a given class, and WaffleCLIP offers no interpretability at all. This interpretability can enable model debugging, as erroneous predictions can be traced back to attributes that either do not match the intended class (i.e. LLM mistake) or cannot be recognized well (i.e. VLM mistake). At a high level, while standard zero-shot classification is a complete black box, the LLM-inferred attributes of our method provide *interpretable intermediate outputs*, increasing the transparency of the system overall. Moreover, our inference strategy results in concise explanations, which have more utility than explanations that are too long for a human to digest [324].

11.5.2.2 Anticipating and Articulating Potential Failures

In addition to explaining each inference, the interpretable intermediate outputs of our pipeline also allow for *error anticipation*. Namely, by comparing the inferred attributes for each class, one can anticipate and describe similar subpopulations from different classes, which may correspond to inputs where the model is less effective. For example, for the Living-17 task in the Breeds datasets, the LLM lists *gibbon* as a kind of both the `ape` and `monkey` classes. While gibbons are apes, they are smaller than most apes, which makes them resemble monkeys. Indeed, standard zero-shot accuracy for gibbons is only 14%, where as other apes are classified at an accuracy of 93.5% [2]. In another case, *rug* is listed as a co-occuring object for the `bed` class, when

[2] We can compute this because *gibbon* happened to be a groundtruth subpopulation for the `ape` class, which does not occur in the `bed` and `rug` case.

`rug` itself is another class in the dataset. While anticipating potential failure modes is intuitive for humans, it is challenging to do so *at scale*. By incorporating an auxiliary model (LLM) with interpretable intermediate outputs (inferred attributes), practioners can both more easily audit



Figure 11.5: Instances where our method corrects mistakes of the standard approach. The attributes used in inference also serve as faithful fine-grained explanations. Notably, these samples are atypical, suggesting that inspecting samples where our method and standard classification disagree can enable automatic surfacing of atypical cases, towards better understanding the task at hand.

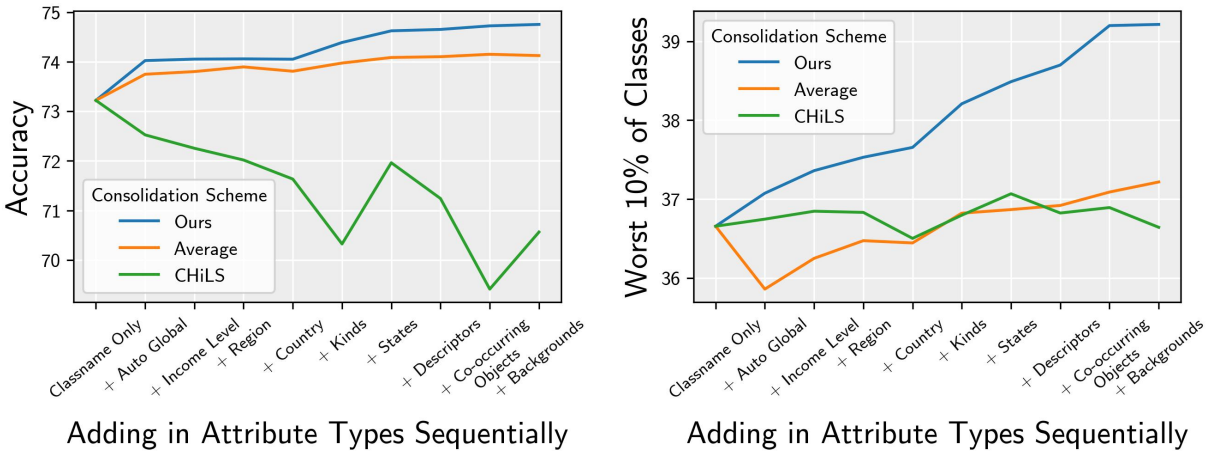


Figure 11.6: Accuracy, overall and for the worst classes, as new types of attributes are added. Performance for our consolidation scheme continuously improves, while it saturates or deteriorates for others.

and verify the zero-shot classification pipeline, and better understand potential challenges with the task of interest. We hope that the greater transparency of our system (importantly, achieved *without compromising on overall accuracy*) can result in increased trust and more responsible use.

11.6 Ablations

We now detail additional ablation studies to shed insight on the source of our method's improvements over existing art and how a practitioner can apply our method with greater control. First, we study how performance varies for both our method and baselines as the number of attributes grows, so to demonstrate the value of our flexible consolidation strategy, specifically for inputs from the hardest classes. Then, we identify a principled trade-off between accuracy overall and on the worst classes using our method, controlled by the hyperparameter k.

11.6.1 Scaling with the Many Axes of Diversity

One source of gains for our method is that we infer attributes of *many* types, while prior works only include one. We argue that our flexible consolidation (of subpopulation similarities to a single class score) also provides improvements over naive averaging or the nonlinear consolidation of CHiLS. To test this, we sequentially add each type of attribute, and inspect performance using the three methods. Figure 11.6 shows our consolidation scales best as more attributes are added, with sizable gains for accuracy over the worst classes. In contrast, performance saturates with averaging, and actually deteriorates with CHiLS. The latter occurs since CHiLS assumes that subpopulations are mutually exclusive, as is the case in hierarchical label sets. When adding attributes along the many axes of diversity, resultant subpopulations overlap, making a zero-shot classification over all subpopulations (as done in CHiLS) unreliable. Averaging is also suboptimal, as the impact of each attribute diminishes as the number of attributes added increases: we see this in the left plot, as accuracy barely increases for the final three added attribute types. Also, samples that are close to only a few subpopulations but far from most (i.e., atypical instances) ultimately receive a low score when all scores are averaged. Thus, while averaging over subpopulations can improve accuracy (to an extent), it is less suited to improving performance on atypical instances than our method. We explore this further in the next section.

11.6.2 Tunable Trade-off between Accuracy Overall and On Worst Classes

Recall that our method consists of computing the similarity of a given test image's embedding to the embedding of numerous (on the order of hundreds) subpopulations per class, before averaging over only the top k similarities, where k is small. Note that when $k = \infty$, our con-

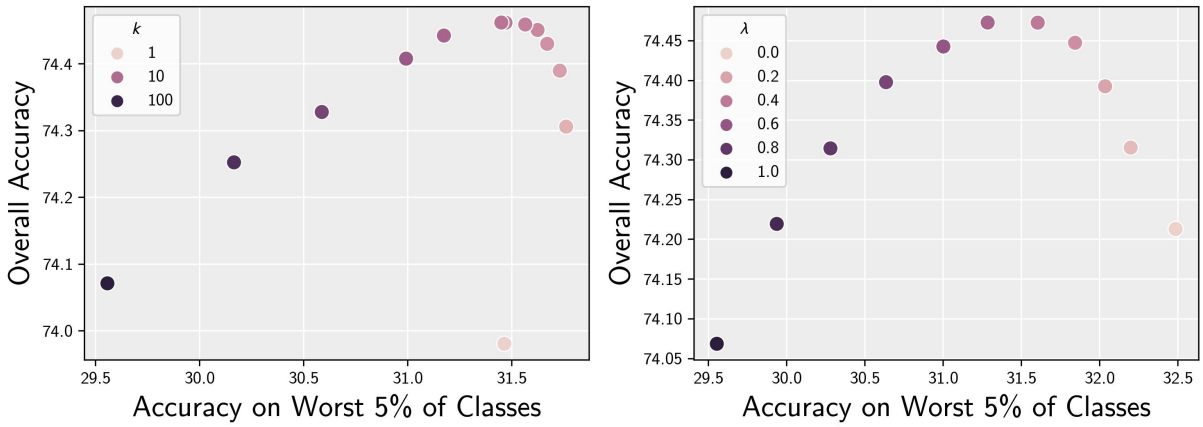


Figure 11.7: **Right:** As k decreases, first, accuracy overall and on the worst classes both increase. Then, overall accuracy begins to decrease while accuracy on the worst classes continues to improve. Thus, we can control this trade-off via k. **Left:** λ, the continuous analog of k, allows for greater control.

solidation reduces to simple averaging over all vectors per class. To shed insight on how our consolidation differs from averaging, we sweep k, while keeping our attribute inference fixed. Additionally, we explore linearly interpolating class scores using our consolidation (top-k) and full averaging via a second hyperparameter λ, so that $\lambda = 0$ results in our method and $\lambda = 1$ is averaging. We jointly sweep λ from 0 to 1 and k from 1 to 128 to pinpoint the way in which our consolidation improves upon averaging.

Figure 11.7 shows overall accuracy vs. accuracy on the worst 5% of classes[3] for both k and λ. The trend is identical for the two parameters: first, both accuracy metrics increase as we move away from full averaging, with much larger gains occurring for the worst classes. Then, accuracy begins to drop, while accuracy on the worst classes continues to improve. To understand this trade-off, consider an instance that has high similarity to one subpopulation embedding for a class, and low similarity to all others. In the $k = 1$ case, this instance is given a high score for the class. This can benefit atypical instances of the class, as they may be visually dissimilar from

[3]We observe the same trade-off when inspecting the worst 10% and 20% of classes.

most other instances (recall the *Arctic* `fox`). However, this can introduce errors, as the correct prediction for an instance mostly close to embeddings from its true class can be flipped with the presence of just one highly similar (perhaps unreliable) subpopulation embedding from a different class. Thus, lower choices of k may benefit more atypical instances, leading to improved accuracy on worst classes (which are most diverse; see 11.3.1), potentially at the cost of overall accuracy. With this insight, practitioners can choose how to tune our method based on their end goals. Also, since λ is continuous, it offers closer control of this tradeoff: indeed, accuracy on the worst classes can be improved by a larger margin when varying λ, and varying k and λ together can lead to best numbers for both metrics.[4]

11.7 Limitations

On utilizing auxiliary models. Our method adds an LLM into the zero-shot classification pipeline, which can increase computational cost and introduce a source of error. We note that the added compute for inferring attributes is only done once per task, so the asymptotic cost per inference only differs marginally (due to computing similarities to more vectors per class, which is a very fast operation) compared to the standard approach. To inspect the reliability of LLM outputs, we manually verify 300 randomly selected LLM outputs. We find only 2.7% of responses are uninformative[5], and none of the 300 to be inaccurate. Moreover, our flexible consolidation scheme offers a kind of robustness to irrelevant LLM outputs: Recall, only the similarity to a small number of subpopulations per class contribute to each logit. Thus, irrelevant (i.e. not appearing in the data) subpopulations are effectively ignored and do not affect the logit.

[4]To be true to the zero-shot setting, no tuning was done to obtain the results in 11.5.1. We tried two reasonably small values for k (8 and 16), observed similar results, and went with $k = 16$, which was marginally better.

[5]All these were found in responses to the 'descriptors' query, with responses like 'medium size' for class `dog`.

However, we still know that LLMs are capable of providing inaccurate outputs, and even detected one such instance (the *gibbon* example from section 11.5.2.2). We find the automatic detection of unreliable LLM outputs to be an interesting avenue for future work, both to improve accuracy and gain insight of potential complexities in the given task.

VLMs are not always reliable. Our work assumes that VLMs are capable of recognizing subpopulations within a class when named. While this is often true, VLMs can still fail, especially for composite concepts. We aim to keep our subpopulations simple, ascribing only one attribute to each. Nonetheless, it is currently not possible to know apriori whether a VLM can recognize a subpopulation in a zero-shot manner. We hope more work on uncertainty estimation can enrich our method, by way of automatically flagging and removing subpopulations that the VLM will not be able to reliably detect.

11.8 Conclusion

To represent classes with diverse instances, which can come in many forms, one vector per class may not be enough. Moreover, VLMs have amazing abilities that are restricted when we only use one vector per class. Thus, instead of ignoring intra-class diversity, we *embrace* it, by explicitly inferring and encoding as much of it as we can. We propose a simple nonlinear consolidation scheme that flexibly attends to subpopulations present in an image while ignoring those that are irrelevant. We find that our method consistently matches or improves over strong baselines, and careful ablations indicate that our method's gains come from improving performance on the hardest classes and subpopulations. Thus, embracing diversity can help reduce performance disparities, including on real-world fairness benchmarks, towards models that work

well *for all*. Our approach allows powerful models to work together in a transparent way via intermediate interpretable outputs, facilitating inferences with explanations, as well as greater tools to understand and debug potential failures. We hope our work spurs further curiosity around how existing paradigms may limit the capabilities of our modern models, towards the development of new AI systems that overcome the fairness and transparency limitations of today.

11.9 Appendix: A deeper understanding

11.9.1 When can we cram an entire class in one vector, and when can we not?

Arguably, diversity within classes is unavoidable, as two instances can vary in numerous ways (discussed further in Section 11.4.1). How then, have classifiers enjoyed success under the one-vector-per-class paradigm, despite its tension with intra-class diversity? First, we note these performance disparities are often obfuscated in metrics like overall accuracy; indeed, the supervised classifiers studied above each achieve impressive overall accuracies. Nonetheless, the tension can be somewhat resolved if (i) one learns embeddings that reduce the diversity that is present in input space, and/or (ii) the single vector learned per class contains features that are unique to the class and present across class instances, despite intra-class variance that persists in the embedding space. We expand on these below.

11.9.1.1 Ideal conditions for the one-vector-one-class paradigm

Most modern vision classifiers consist of a deep feature encoder, mapping images to a rich embedding space, followed by a linear classification head, mapping embeddings to class logits. The linear classification head consists of a single vector (and a scalar bias) per class. A linear

classification head is accurate if, for any instance from the i^{th} class, the activation on the i^{th} class vector must be higher than the activation for any other class vector. We express this mathematically below, with $\mathbf{x}$ denoting the embedding of an image from class i, and $\mathbf{c_i}, \mathbf{c_j}$ denoting vectors in the classification head.

$$\forall \mathbf{x} \in \mathcal{C}_i, \forall j \neq i, \text{ we require that } \mathbf{x} \cdot \mathbf{c_i} - \mathbf{x} \cdot \mathbf{c_j} > 0 \tag{11.1}$$

$$\text{Note that } \mathbf{x} \cdot \mathbf{c_i} - \mathbf{x} = \mathbf{x} \cdot (\mathbf{c_i} - \mathbf{c_j}) = \|\mathbf{x}\| \|\mathbf{c_i} - \mathbf{c_j}\| \cos(\mathbf{x}, \mathbf{c_i} - \mathbf{c_j}) \tag{11.2}$$

$$\text{Thus, } \forall \mathbf{x} \in \mathcal{C}_i, \forall j \neq i, \text{ we require that } \cos(\mathbf{x}, \mathbf{c_i} - \mathbf{c_j}) > 0 \tag{11.3}$$

The last step arises because norm is always non-negative. Now, let us focus on different components of this required condition (by definition) for an accurate one-vector-per-class classification head. First, the single vector $\mathbf{c_i}$ must contain contain a set of features that are *unique* to that class. That is, these features remain when considering the residual $\mathbf{c_i} - \mathbf{c_j}$ for any $i \neq j$. Secondly, the unique features that discriminate the class from all others must also be aligned with every instance of the class. In other terms, these unique features must be *invariant* to any diversity within the class. Also, note that the quantity we expand upon above is simply the margin for classification. In the ideal case, this margin would be maximized.

11.9.1.2 Class-supervised training is well suited for the one vector per class paradigm, but VLM pretraining is not

In traditional class-label supervised training, the feature encoder is jointly optimized with the classification head to minimize a classification loss. Let us consider how this effects the linear classification head and the feature encoder individually. First, fixing the classification head, we see the supervised objective encourages all embeddings from one class to be drawn close to their respective single vector, and consequently, close to one another. In other words, invariance of embeddings within a class is promoted. Next, with the feature encoder fixed, classification head vectors align with embeddings within their class and de-align with embeddings from outside their class. Thus, the classification head vectors are optimized to solely contain the features unique to their class embeddings. Therefore, training with traditional class-label supervision directly promotes the invariance and uniqueness properties required for the success of the one-vector-per-class paradigm.

On the other hand, VLMs are optimized with markedly different objectives. Many VLMs employ contrastive image-text matching, in which negative examples are far weaker and classes are no longer defined; in some ways, the training is analogous to optimizing a classification task with an infinite number of classes. Indeed, two instances that belong to the same class in a downstream task may have embeddings pushed apart during VLM pretraining, directly going against the aforementioned notion of class-wise invariance. Other common VLM objectives like captioning or question answering promote the descriptiveness of the embedding. Thus, instead of honing in on unique features, embeddings are likely to describe as much as possible. We note that having maximally descriptive embeddings is typically a good thing, as it allows for re-use

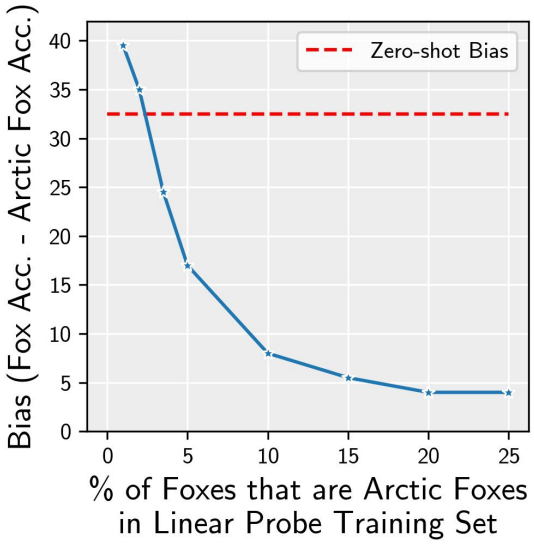


Figure 11.8: Arctic Fox bias is amplified in zero-shot classifier vs. to supervised linear probes.

of the same feature encoder for many downstream tasks, as is done in linear probing with self-supervised encoders. The key caveat is that in those cases, the linear classification head is still exposed to instances from all classes, and thus, each classification head vector can learn to align only with the unique features for its class. In contrast, in the zero-shot setting, the classification head vectors are obtained independently of one another via embedding the names of classes via the text encoder, and thus, it is unreasonable to expect that these vectors satisfy the uniqueness condition.

11.9.1.3 Arctic Fox Case Study: Bias can be amplified when using one vector per class paradigm for zero-shot classification

Staying in the one-vector-per-class setting, we now compare class vectors obtained directly in a zero-shot manner to those obtained with supervision. Specifically, we focus on the `Arctic Fox` bias, shown in Figure 11.2. We train a linear classification head over fixed CLIP embeddings used a skewed training set that under-represents `Arctic foxes` in the training set. We find

that the bias of the zero-shot vector is on par with having only 3% of the training images in the `fox` class be `Arctic foxes` in the supervised setting, suggesting that limitations of the one vector per class paradigm may be exacerbated in the zero-shot setting.

11.9.2 One final Trade-off

In section 11.6.2, we showed two hyperparameters that could trade overall accuracy for accuracy over the worst classes. We now present one more, along with a *theoretical* explanation. Throughout the paper, we consider 'averaging' to mean computing similarities to multiple vectors and then averaging those similarities; this is how DCLIP and WaffleCLIP average, and will refer to this as **Average Sims**. However, averaging over prompts as done in originally in CLIP consists of averaging vectors first and then computing similarity to one average vector; we call this **Average Vecs**. The difference is subtle: in the latter case, an additional normalization occurs when cosine similarity is taken.

We now show theoretically that when all embeddings are normalized (i.e. for CLIP), **Average Vecs** simply rescales the class score yielded by **Average Sims** by a factor that measures how *diffuse* the vectors for the class are. Let x be an image embedding and $\{v_1, v_2, \ldots, v_k\}$ be subpopulation vectors for a given class. We assume all vectors are normalized to the hypersphere, as is the case for CLIP. That is, $\|v_i\| = 1$ for all i and $\|x\| = 1$. Let $\overline{v} := \frac{1}{k}\sum_{i=1}^{k} v_i$ denote the average vector. We compute the class score for **Average Vecs** below.

$$\textbf{Average Vecs} = \cos(x, \overline{v}) = \frac{x \cdot \overline{v}}{\|x\|\|\overline{v}\|} = \frac{x \cdot \frac{1}{k}\sum_{i=1}^{k} v_i}{\|\overline{v}\|} = \frac{\frac{1}{k}\sum_{i=1}^{k} x \cdot v_i}{\|\overline{v}\|}$$
$$= \frac{\frac{1}{k}\sum_{i=1}^{k} \cos(x, v_i)}{\|\overline{v}\|} = \frac{\textbf{Average Sims}}{\|\overline{v}\|}$$

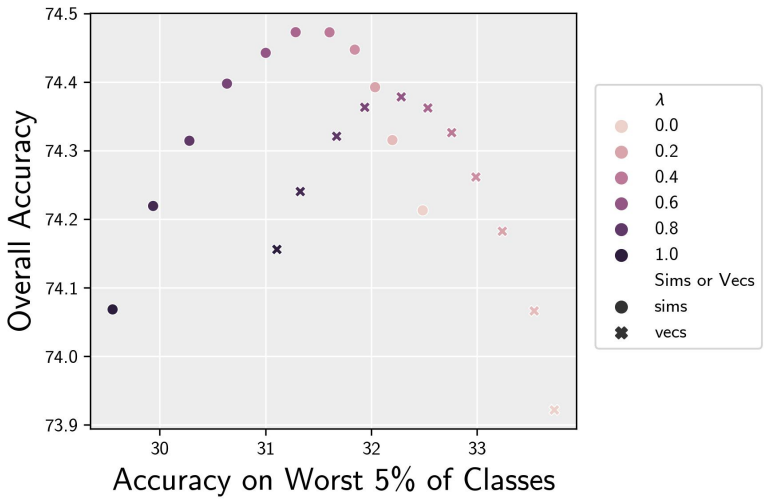


Figure 11.9: Averaging subpopulation vectors before computing similarity to an image embedding proves to be another way to trade overall accuracy for accuracy on the worst classes. That is, when we first compute similarity to each subpopulation and then average, we obtain higher overall accuracy but lower accuracy on the worst classes, compared to when we first average subpopulation vectors and then compute the similarity to the average vector.

To get from line 1 to 2, we utilize the fact that cosine similarity is equivalent to the dot product when both arguments are unit norm. Let us now consider what this result entails. The denominator is the norm of the average vector. This quantity is always between 0 and 1. It is lowest when the vectors are most diffuse. Thus, the class score obtained by **Average Sims** is scaled up to obtain the score for **Average Vecs** by more when the vectors are diffuse. In other words, averaging the vectors first implicitly upweights vectors corresponding to diverse subpopulations.

Based on this simple theory, we would expect the most classes with high diversity to have higher accuracy under **Average Vecs** compared to **Average Sims**, as their class scores are inflated more than the less diverse classes. The effect on overall accuracy, however, is not perfectly clear. To inspect this, we perform the same sweep over k and λ as in section 11.6.2, except now we additionally try replacing all similarity averaging with vector averaging. Figure 11.9 shows the

results. We average away k for clarity. Indeed, averaging over vectors improves accuracy on the worst classes. For high values of $\lambda = 1$, we see averaging vectors also slightly improves overall accuracy. However, in the vast majority of values for λ, overall accuracy is hurt by averaging vectors instead of similarities. We hope this analysis provides insight as to the precise effect of averaging similarities or vectors, which may be relevant to others who wish to explore going beyond one vector per class.

Chapter 12: Rethinking Artistic Copyright Infringements in the Era of Text-to-Image Generative Models

Finally, we study a problem that has direct human impact and is fundamentally interdisciplinary: copyright of individual artistic styles. This chapter details a paper accepted to ICLR '25 co-authored by Sriram Balasubramanian, Samyadeep Basu, Priyatham Kattakinda, Atoosa Chegini, Robert Brauneis and Soheil Feizi [325]. Notably, Robert Brauneis is a seasoned veteran in the field of copyright law. Our challenge in this work was to devise a technical solution that primarily non-technical audiences – lawyers, judges, juries, artists – could trust. Namely, transparency is crucial here. Thus, we build off the lessons from prior chapters to develop a practical tool to deliver immediate positive societal impact.

The advent of text-to-image generative models has led artists to worry that their individual styles may be copied, creating a pressing need to reconsider the lack of protection for artistic styles under copyright law. This requires answering challenging questions, like what defines style and what constitutes style infringment. In this work, we build on prior legal scholarship to develop an automatic and interpretable framework to *quantitatively* assess style infringement. Our methods hinge on a simple logical argument: if an artist's works can consistently be recognized as their own, then they have a unique style. Based on this argument, we introduce `ArtSavant`, a practical (i.e., efficient and easy to understand) tool to (i) determine the unique style of an artist

by comparing it to a reference corpus of works from hundreds of artists, and (ii) recognize if the identified style reappears in generated images. We then apply `ArtSavant` in an empirical study to quantify the prevalence of artistic style copying across 3 popular text-to-image generative models, finding that under simple prompting, 20% of 372 prolific artists studied appear to have their styles be at risk of copying by today's generative models. Our findings show that prior legal arguments can be operationalized in quantitative ways, towards more nuanced examination of the issue of artistic style infringements.

12.1 Introduction

Currently, US copyright law offers no protection for artistic styles, likely due to challenges in defining style infringement and a prior lack of necessity. However, with highly capable text-to-image generative models transforming the art landscape, there have been increasingly more calls to reconsider this lack of protection. Namely, many fear that models like Stable Diffusion, Imagen, Mid-Journey, and DeepFloyd [243, 244, 326, 327] may make replication of an artist's unique style as simple as providing an adequate prompt, potentially inundating the market with imitations that devalue the original artist's work and threaten their livelihood. This matter has gained widespread attention, as it is fundamentally interdisciplinary (engaging legal, artistic, and technical communities), and has serious material and human consequences. In fact, multiple legal cases regarding artistic copyright are ongoing [328, 329], and [330] has already called for new provisions in copyright law to protect artistic style. Thus, with generative AI creating a pressing need to answer previously unsettled questions, it is now critical for us to rethink artistic copyright.

We begin by surveying existing copyright law around art, finding that it has historically

relied heavily on qualitative judgments, such as the 'substantial similarity test' [331]. The subjective nature of existing law has led to judgments that vary from case-to-case (e.g. on if characters are protected [332, 333, 334] or not [335]), which will prove problematic when dealing with AI-powered mimicry at unprecedented scale. An automatic quantitative approach could help make judgments quicker and more consistent. However, any technical solution *must* also be easy to understand and based in legal literature, as the audience whose decisions we intend to assist[1] is largely non-technical (e.g. judges, juries, and artists). Thus, we set out to develop an intuitive, automatic, and legally-grounded manner to quantitatively argue artistic style infringement.

Our work is structured around key questions we identify that, when answered, can form the basis of an argument for artistic style infringement. Namely, given an artist, **i.** do they have a unique style? If so, **ii.** to what degree does it appear in generated art? And lastly, **iii.** in what precise way (with respect to specific artistic elements present in both the original and generated art) is the style copied? Inspired by legal arguments presented by [336] and [337], we first frame artistic style as characterized by a set of elements that co-occur frequently across an artist's *body of work*, where each element may not be protectable on its own, but together, they could represent an unique stylistic *signature*.

To prove the *uniqueness* of an artistic style, we propose a simple logical argument, which serves as our core ideological contribution. Namely, if an artist's works are consistently recognized as their own, that artist must have some unique style, with which their work can consistently be classified back to them (instead of hundreds of other potential creators). Therefore, the task of showing the existence and uniqueness of artistic styles can be studied *from the lens of classification* – something deep networks are particularly adept at doing. To perform this classification, we

[1] We absolutely do not wish to replace humans in this nuanced decision making, but instead to help them.

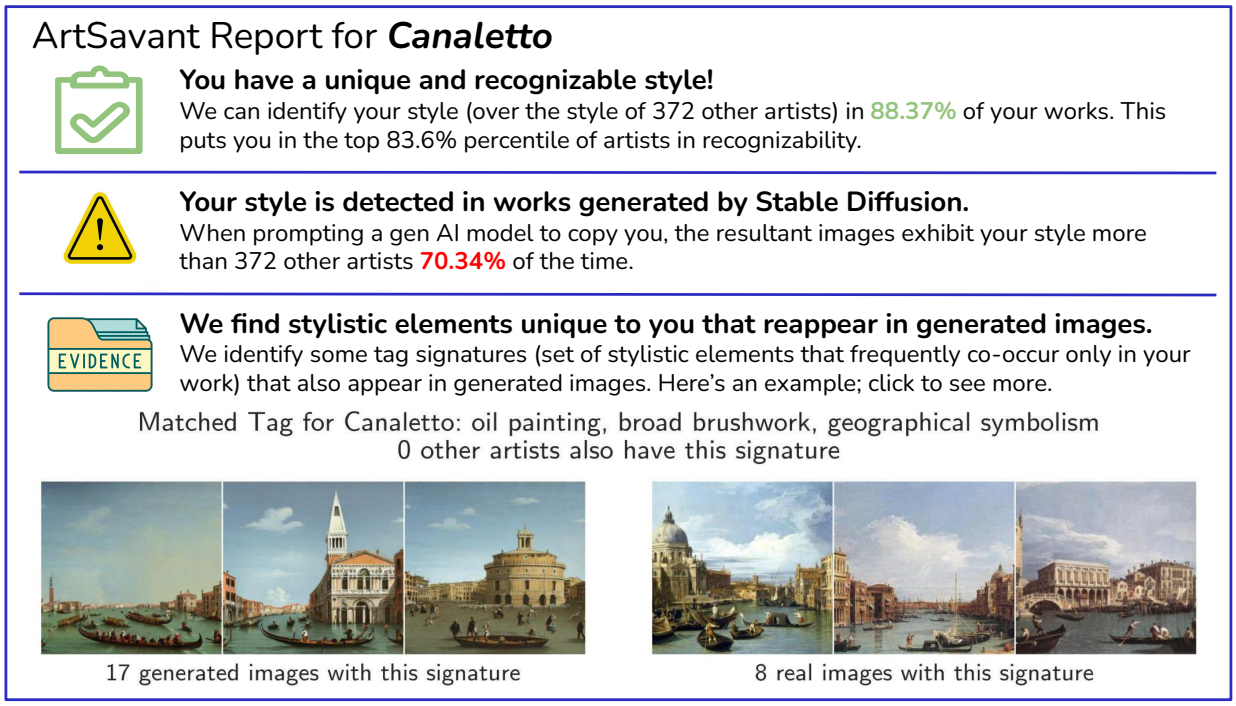


Figure 12.1: Our primary contribution is an interpretable and quantitative framework, rooted in legal scholarship, for arguing style infringement from the perspective of classification. Given artworks by Canaletto, our tool, `ArtSavant`, automatically identifies a unique style and recognizes said style in generated art, summarizing its findings in an easy to understand yet quantitative report.

curate a reference dataset of artworks from 372 artists, and implement two classification methods, taking holistic and analytic approaches.

Current copyright law dictates that similarity between artworks must be evaluated in *analytic* and *holistic* terms [331, 338] (see § 12.2). For example, we could *analyze* Vincent Van Gogh's style as comprised of expressive wavy lines, bright unblended coloring, post-impressionism, choppy textured brushwork, etc, or we could make an intuitive, *holistic* judgement: e.g. in Figure 12.3, we can tell the look and feel of the generated images capture Van Gogh's style, even without articulating the shared stylistic elements. In order to make these notions more concrete and quantitative, we develop two complementary methods corresponding to holistic and analytic ways of evaluating style similarity.

The first method – *DeepMatch* – is simply a neural network classifier, with which we show that for 89.2% of artists, their held-out works can consistently (i.e. over half the time) be mapped back to them (over 371 other artists). This represent a significant empirical finding, as it quantitatively shows that **unique artistic styles exist for an overwhelming majority of artists** we study – a necessary precondition for adding legal protections for artistic style. DeepMatch can be thought of as capturing *neural signatures* for each artist. Namely, the classification head vector consists of a combination of neural features that encode that artist's style. As these are not interpretable, we complement the *holistic* DeepMatch with *TagMatch*, a novel, inherently interpretable, attributable, *analytic* approach.

TagMatch maps a collection of works to an artist by first assigning diverse stylistic tags in a zero-shot manner to each work, and then efficiently searching over the space of tag compositions to surface unique *tag signatures*. We show empirically that searching over tag compositions is critical, as no tag on its own is unique to an artist – recall this idea was described by legal scholars [336] and [337]; in our work, we make this quantitative. Further, importantly, **TagMatch allows for articulating the specific combination of elements that uniquely appear in artist's works and in (infringing) generated work**. Such interpretable output can be particularly useful for the legal settings where arguments around artistic copying ultimately are to be made.

We package these methods in `ArtSavant`, a tool that for any artist can produce a report like Figure 12.1 in minutes. `ArtSavant` shows our logical framework to argue style infringement is practically feasible, automatically generating a quantitative assessment of style copying in a way that can be understood by the broad set of relevant stakeholders. Finally, we use `ArtSavant` to provide a quantitative snapshot of the prevalence of style copying with today's generative models under simple prompting. After generating images in the style of artists from

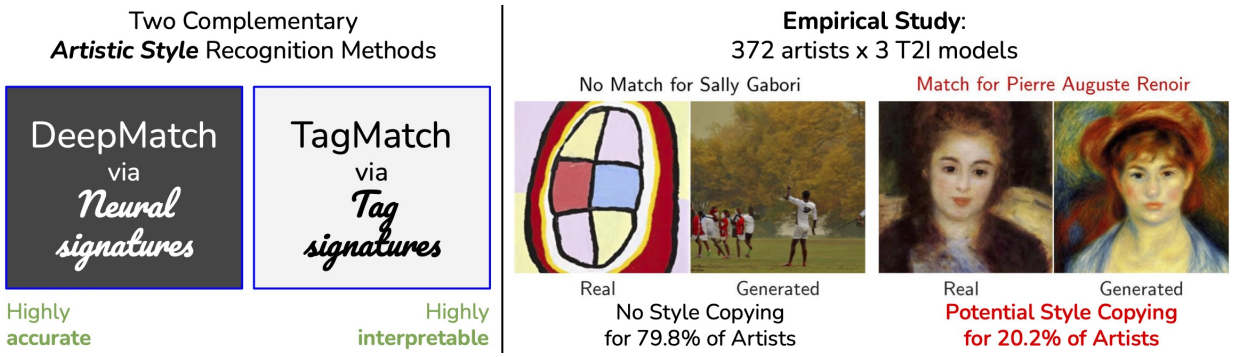


Figure 12.2: We define artistic style as a set of elements (or signature) that appear frequently over a body of work, and reduce the problem of style copy detection to classification of *sets* of images to artists. (**left**) We offer proof-of-concept via two ways to recognize artistic styles over image *set*, including a novel inherently interpretable and attributable tag-based method. (**right**) In an empirical study of 372 prolific artists, we find generative models potentially copy artistic styles for 20.2% of these artists.

our dataset with 3 popular text-to-image models and using `ArtSavant` to assess copying, we find 20% of the artists we study to be at risk, suggesting style copying may indeed by possible with today's generative AI.

In summary, we make the following contributions:

- We develop an automatic, interpretable, and **legally-grounded manner to quantitatively argue artistic style infringement**, leveraging a simple classification argument.
- We introduce `ArtSavant`, **a practical tool to quantify and communicate style copying for any artist**, consisting of a reference dataset of artworks from 372 prolific artists, and two complementary methods, including a novel, highly interpretable and attributable one. Notably, our two methods directly relate to the principles of "analytic" and "holistic" style similarity used in the substantial similarity test in existing copyright law.
- With `ArtSavant`, we perform **a large-scale empirical study** to measure style copying across 3 popular text-to-image generative models, finding that generated images (using

simple prompting) from 20% of the artists examined appear to be at risk of style copying.

12.2 Related Works

Overview of copyright legal literature: Currently, there is an ongoing debate within legal scholarship[2] regarding whether artistic style can and should be protected under copyright law. Some [339] conclude, from a traditional interpretation of copyright law, that style is wholly outside the purview of copyright law as copyright is only concerned with protection against copying *individual* artworks. However, many courts have recognized protection for characters that are developed over multiple works, from Sherlock Holmes [334] to James Bond [333] and the Batmobile [332]. Other scholars [336, 337] have argued that artistic style, separate from any particular artwork, constitutes a distinct intellectual contribution which has commercial and aesthetic value. In particular, [336] suggests that style can be defined as "a holistic attribute of a work, or a group of works, that comprises a constellation of expressive choices. These expressive choices might be unprotectable individually, but in combination, they may constitute protectable expression." Additionally, Sobel emphasizes the challenge courts face when applying the *substantial similarity test* to art – they "must simultaneously dissect images into their constituent elements — a task judges claim they are unable to do— while also assessing works' aesthetic appeal holistically and intuitively". If indeed style may come under copyright protection in the future, a substantial similarity test for art style would be even more challenging to implement and apply than one for individual artworks. In our work, we aim to develop a tool which can help courts determine the extent of style similarity from the lens of the substantial similarity test applied to artistic style.

[2]We encourage readers to review Appendix 12.7.1 for a detailed but brief history of copyright law, so to understand the broader context for our work and the legal challenges / ambiguities surrounding it.

Critically, we do not seek to replace human decision makers with an automated analog, but instead to leverage our tool to provide quantitative evidence to assist humans in making nuanced qualitative judgments.

Current technical solutions: The rapid advance of image generative models has made the possibility of mimicking artists' personal styles a topic of discussion in the literature [340]. Some works describe ways to either detect direct image copying in generated images, or to foil any future copying attempts by imperceptibly altering the artists' works to prevent effective training by the generative models. These include techniques like adding imperceptible watermarks to copyrighted artworks [341, 342, 343], and crafting "un-learnable" examples on which models struggle to learn the style-relevant information [344, 345, 346]. Others have suggested methods to mitigate this issue from the model owner's perspective - to either de-duplicate the dataset before training [347, 348, 349], or to remove concepts from the model after training ("unlearning") [350, 351, 352]. Methods like [347, 348, 349] are also more focused on analyzing direct image copying from the training data, and thus may not be applicable to preventing style copying.

Shortcomings of current methods from a legal perspective: None of these works tackle the problem of *detecting* potentially copied art *styles* in generated art, especially in a manner which may be relevant to legal standards of copyright infringement. According to current US legal standards [353], an artwork has to meet the "substantial similarity" test for it to be infringing on copyright. This similarity has to be established on *analytic* and *holistic* terms [331, 338]. Analytic here refers to explaining an artwork by breaking it down into its constituents using a concrete and objective technical vocabulary, while holistic refers to the overall "look and feel" of the artwork. So to be relevant to the legal community (who ultimately decides on alleged cases of style copying), we design our tool to reflect this dichotomy in its working, while also empha-

sizing ease of use and interpretability, to make our tool practically useful for a concerned artist hoping to protect themselves. These priorities manifest in our reformulation of detecting style copying as classification in §12.4. But first, we discuss limitations in applying the typical copy detection approach to artistic styles.

12.3 Motivation: Image-wise similarity may be limited for Style Copying

A prevailing approach to investigating copying involves representing images in a deep embedding space via models like SSCD [354] or DINO [355], and computing image-to-image similarities across generated and real images. Such an approach has been employed by [347, 348, 349] to show that generative models can (though rarely do) create exact replicas of training images. Inspired by these results and the consequent concerns from artists, we first explore if generative models can recreate famous artworks, e.g., by Vincent Van Gogh. Specifically, we generate images by prompting "{*artwork title*} by Vincent Van Gogh" for 1500 Van Gogh works, and compute the DINO similarity between pairs of a real and corresponding generated image. Figure 12.3 visualizes the distribution of similarities, as well as examples at each similarity level. We find that the vast majority of similarities are lower than 0.75, which amounts to pairs that are far from duplicates. However, even when the generated image differs significantly from the source real image, certain stylistic elements associated with Van Gogh seem to appear consistently in the generated works. Thus, **while instance-wise copying of artwork appears rare for even the ultra famous Van Gogh, style copying may require going beyond image-to-image comparisons**, as artists may still have their personal styles, developed over a long career/many artworks and at significant personal cost, infringed upon in ways that searching for exact replicas would miss. A

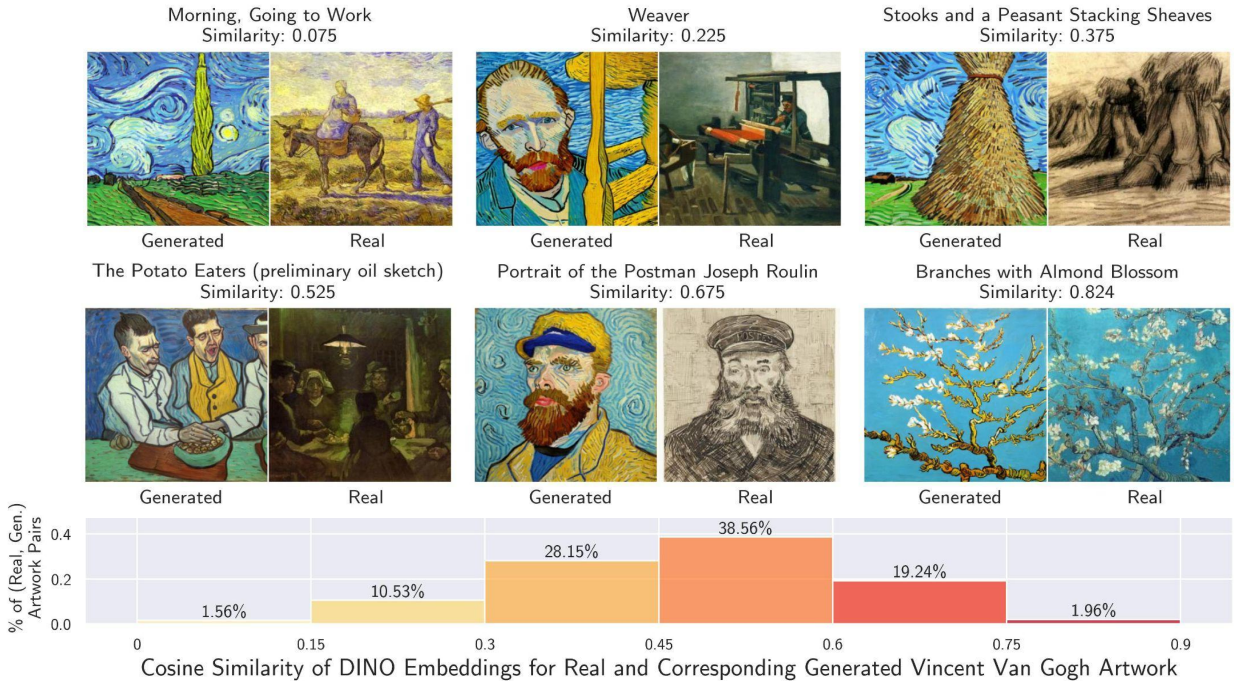


Figure 12.3: Example generations from Stable Diffusion 2 when prompted to produce specific paintings by Vincent Van Gogh, along with the histogram of similarities between the generated image and corresponding real image. Even for a famous artist like Van Gogh, generative models rarely produce near-exact duplicates. However, Van Gogh's *style* appears consistently, even when similarity is low.

recent work finetunes embeddings so that cosine similarity better proxies style similarity [356], though even in this case, the utility of such a tool in court is limited by its lack of interpretability.

12.4 Reformulating Artistic Style Copying as Classification over Image Sets

Having established that style is comprised over a body of work (instead of a single image) and that copy detection must be interpretable to hold weight in court, we now present an alternate framework for arguing style infringement, with the following intuition: if an artist's work can consistently be distinguished from that of other artists, then there must exist something unique that is present across that artist's portfolio. Thus, we can use classification over image sets (i.e. bodies of work) to demonstrate a unique style exists given an artist. Then, style infringement can

be argued by showing the copied artist can again be predicted (over many others) given a set of generated works. We now detail DeepMatch and TagMatch, two complementary methods (w.r.t. accuracy and interpretability) that classify artistic styles over image sets, in holistic and analytic manners respectively.

A necessary preliminary: WikiArt Dataset. To distinguish one artist's style from that of others, we need a corpus of artistic styles (i.e. portfolios from many artists) to compare against. To this end, we curate a dataset $\mathcal{D}$ consisting of artworks from WikiArt [3] (like others [357, 358]) to serve as (i) a reference set of artistic styles, (ii) a validation set of real art to show (most) artists have unique styles and our methods can recognize them on held-out sets of their works, and (iii) a test-bed to explore if text-to-image models replicate the styles of the artists in our dataset in their generated images. We include $\sim$91k artworks from 372 artists $\mathcal{A}$ spanning diverse eras and art movements, including any artist with at least 100 works on WikiArt. Each work is labeled with its genre (e.g., *landscape*) and style (e.g., *Impressionism*), though we primarily use the artist and title labels. We provide an easy-to-execute script to enable others to scrape newer versions of this dataset if desired. We now detail DeepMatch and TagMatch, which each compare a test set of images to our reference corpus.

12.4.1 DeepMatch: Black-Box Detector

DeepMatch consists of a light-weight artist classifier[4] (on images) and a majority voting aggregation scheme to obatin one prediction for a *set* of images. Majority voting requires that at least half the images in a test set $\hat{D}$ are predicted to an artist a for DeepMatch to predict a,

[3] https://www.wikiart.org/; **note that we only include Public domain or fair use images**.

[4] Others have trained art classifiers [358, 359, 360], but they do not operationalize them for style infringement.

allowing for abstention in case no specific style is recognized with sufficient confidence. For our classifier, we train a two layer MLP on top of embeddings from a frozen CLIP ViT-B\16 vision encoder [13], using a train split containing 80% of our dataset. We employ weighted sampling to account for class imbalance. Since we utilize frozen embeddings, training takes only a few minutes on one RTX2080 GPU. Thus, a new artist could easily retrain a detector to include their works (and thus encode their artistic style).

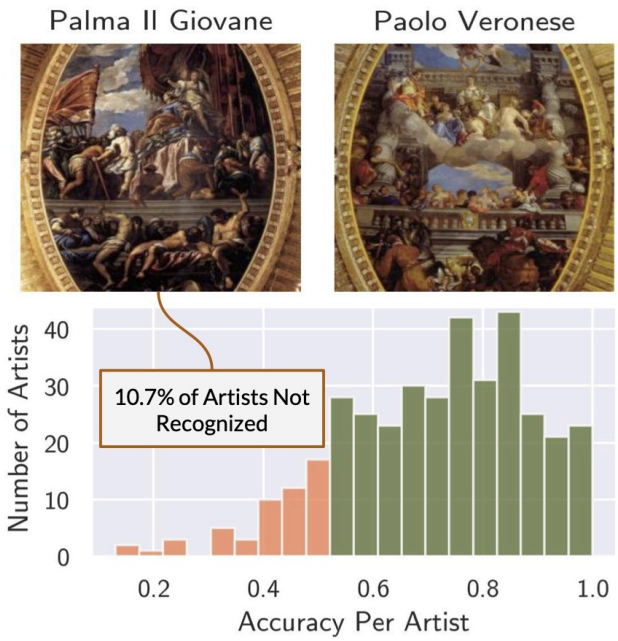


Figure 12.4: DeepMatch on held-out real art: 89.3% of artists can be recognized. The remaining 10.7% of artists have very similar styles to other artists: e.g., Palma Il Giovane's work differs marginally from other Italian renaissance painters.

Validation of the Detector. We apply DeepMatch on the held-out test split of our dataset and observe that the image-wise classifier attains 72.8% accuracy per image over 372 artists. When aggregating image-wise predictions via majority vote, 89.3% of artists are matched, validating our method, and **offering strong evidence towards the existence of unique artistic styles**. Specifically, neural classifiers capture unique and frequently co-occurring characteristics of the artists in their embedding space, which can be thought of as 'holistic' *neural signatures*. Figure 12.4 shows the distributions of image-wise accuracies per artist, shading correctly matched artists (green). We also present an image from one of the few artists who's style is not matched by DeepMatch, along with an image from a similar artist. Notice that the style of two artists can be extremely similar (see Appendix 12.7.2.1), making the existence of unique artistic styles for the vast majority of artists considered

(by way of neural signatures) a non-trivial observation. **This empirical finding also has legal significance, as it is necessary for an artist to demonstrate that they have a unique style before they could allege that their style is being infringed upon.**

12.4.2 Interpretable Artistic Signatures

Now we provide an analytic complement to DeepMatch's holistic approach. Namely, we seek to articulate the elements that comprise an artist's unique style. We do so by tagging images with descriptors (called atomic tags) drawn from a vocabulary of stylistic elements. Then, we *compose* tags efficiently to go from atomic tags that are common across artists to longer tag compositions that are unique to each artist (i.e. *tag signatures*). We detail these steps now, before explaining how tag signatures can be used to classify an image set to an artist in the following section.



Figure 12.5: Example atomic tags assigned via our proposed CLIP-based zero-shot method. We perform selective multilabel classification along various aspects of art (e.g. medium, colors, shapes, etc), so that atomic tags span diverse categories. Details in section 12.4.2.

Zero-shot Art Tagging. We utilize the zero-shot open-vocabulary recognition abilities of CLIP to tag images with descriptors of stylistic elements. First, we construct a concept vocabulary $\mathcal{V}$ with help from LLMs. Namely, we prompt Vicuna-13b and ChatGPT to generate a dictionary of concepts along various aspects of art. We manually consolidate and amend the

concept dictionary, resulting in a vocabulary of 260 concepts over 16 aspects.

To assign concepts to images, **we a design a novel zero-shot scheme that consists of selective multilabel classification per-aspect**. Namely, for an image, we compute CLIP similarities to all concepts, and normalize similarities *within each aspect*. Then, we only assign a concept its normalized similarity (i.e. z-score) exceeds a threshold of 1.75. This means that a concept is only assigned for an aspect if the image is substantially more similar to this concept than other concepts describing the same aspect. Classifying per-aspect allows for a diversity of descriptors to emerge, while global thresholding results in a biased tag description, as concepts for certain aspects (e.g. subject matter) consistently have higher CLIP similarity than those for more nuanced aspects (e.g. brushwork). We call the assigned concepts *atomic tags*; figure 12.5 shows atomic tags assigned for a few examples.

Validation of Quality of Tags Using Human-Study. We validate the effectiveness of our tagging via a human-study involving MTurk workers. In particular, given an image of an artwork and an assigned atomic tag $v_{predict}$ from the vocabulary $\mathcal{V}$ – MTurk workers are asked "*Does the term $v_{predict}$ match (i.e. the concept $v_{predict}$ present) the artwork below?* ". The workers are then asked to select between $\{$Yes, No, Unsure$\}$. We collect responses for 1000 images with 3 annotators each. We find that in only 17% cases, a majority of workers disagree with the provided tag, suggesting our tagging results in a low false positive rate. We also observe all three annotators agree in only 51% of cases, reflecting that describing artistic style can be subjective. While our tagging is not perfect, it is a deterministic and automatic method of articulating artistic style elements, and it will improve as underlying VLMs improve too.

Tag Composition for Artists. Using the atomic tags in the artwork specific vocabulary $\mathcal{V}$, we now present an iterative algorithm to obtain a set of *tag signatures* $\mathcal{S}_a$ for each artist $a \in \mathcal{A}$,

which consist of multiple atomic tags that frequently *co-occur* in an artist's works. In particular, our algorithm efficiently searches the space of tag compositions to go from atomic tags to composition of tags which become more unique as the length of the tag composition grows. For e.g., while 40% of the artists may use simple colors, *only* 15% may use both simple colors and impressionism style *in the same works*.

To efficiently search the space of tag compositions per artist $a \in \mathcal{A}$, we first assign a set of tags to each of their images $x \in \mathcal{D}_a$ via the zero-shot *selective multi-label classification* method described above. For each image x, let $\text{tag}(x)$ denote the set of predicted atomic tags. To get atomic tags *for an artist*, we aggregate all atomic tags over images, and keep only the tags occurring in at least 3 works. We denote this aggregate set of atomic tags as the "Common Atomic Tags Per Artist" and denote it as $\mathcal{C}_a$. Then, we iterate through all the images $x \in \mathcal{D}_a$ for a given artist a, to find the intersection $I(x) = \text{tag}(x) \cap \mathcal{C}_a$. We then compute a powerset $\mathcal{P}(I(x))$ of the tags occurring in the intersection $I(x)$ and increment the count of each occurrence of the tag composition from the powerset in $\mathcal{S}_a$. Note that the size of $I(x)$ is much smaller than that of $\mathcal{C}_a$, and thus, iterating through $\mathcal{P}(I(x))$ for each image x is much, much faster than iterating through $\mathcal{P}(\mathcal{C}_a)$. Finally, we again filter the tag compositions in $\mathcal{S}_a$, only including those that occur in at least 3 works. We provide the details of this tag composition algorithm in 2.

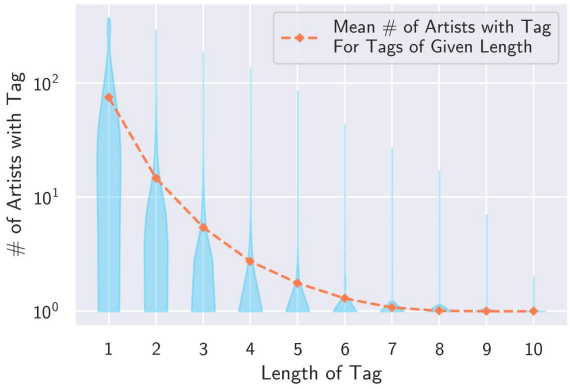


Figure 12.6: Composing atomic tags results in more unique tags, towards artistic *tag signatures*.

Can Unique Artistic Signatures be Articulated? Applying tag composition on the WikiArt

Algorithm 2 Iterative Algorithm to Obtain Tag Composition Per Artist $a \in \mathcal{A}$

Require: $\mathcal{D}_a$ (Images for artist a), $\mathcal{C}_a$ (Common tags for artist a)

$\mathcal{S}_a = \{\}$ ▷ Stores the tag compositions with their associated counts
for $x \in \mathcal{D}_a$ **do**
    $I(x) = \text{tag}(x) \cap \mathcal{C}_a$ ▷ Compute the intersection with common atomic tags
    $\mathcal{P}(I(x)) = \text{ComputePowerSet}(I(x))$ ▷ Compute power-set of the tags
    $\text{UpdateCount}(\mathcal{S}_a, \mathcal{P}(I(x)))$ ▷ Update the count of each tag composition
end for
$\text{Filter}(S_a)$ ▷ Keep tag compositions which occur above a count threshold of 3

dataset, we observe that when the length of a tag (i.e. number of atomic tags that all frequently co-occur) grows, the number of artists that the tag (composition) is common for decreases (fig. 12.6). Eventually, our tag composition algorithm identifies unique tag *signatures* such that *only* one artist frequently produces art containing all atomic tags in the composition. This echoes the previous finding that unique styles exist, with these styles now articulated, towards an interpretable argument of style copying.

12.4.3 TagMatch: Interpretable and Attributable Style Detection

In 12.4.1, we outlined a holistic approach to accurately detect artistic styles. While DeepMatch obtains high accuracy (recognizing styles for 89.3% of artists), the neural signatures it relies upon lack interpretability. For a copyright detection tool to be useful in practice (e.g., to be used as assistive technologies), providing explanations of the classification decisions can tremendously benefit the end-user. To this end, we leverage our efficient tag composition algorithm as defined in 12.4.2 to develop TagMatch - an interpretable classification and attribution method which can effectively classify a set of artworks to an artist, as well provide reasoning behind the classification and example images from both sets that present the matched tag signature. TagMatch follows the intuition of matching a test portfolio to a reference artist who's portfolio

shares the most unique tag signatures. Given a set of N test images $\mathcal{T} = \{x_i\}_{i=1}^{N}$, we first obtain a number of tag compositions for them using our iterative algorithm in 12.4.2. These tag compositions are then compared with the tag compositions of the artists in the reference corpus in order of uniqueness (i.e. we first consider tag signatures present in the test portfolio that occur for the fewest number of reference artists). We can then rank reference artists by how unique the shared tags are with the test portfolio. Notably, TagMatch is fast, taking only about a minute, after caching embeddings of all images.

Validation of TagMatch. We again utilize the test split of our WikiArt Dataset to validate the proposed style detection method. TagMatch predicts the correct artist with top-1 accuracy of 61.6%, with top-5 and top-10 accuracies rising to 82.5% and 88.4% respectively. While less accurate than DeepMatch, the *tag signatures* provided by TagMatch allow for analytic arguments to be made regarding style copying, as the exact tag signatures used in matching can be inspected. Moreover, the subset of images in both the test portfolio and matched reference portfolio can be easily retrieved, offering direct attribution of the method; examples can be seen in the next section, where we match generated images to our reference artists. Overall, we hope TagMatch and DeepMatch can serve as automatic and objective tools to navigate the subtle problem of identifying artistic styles, towards detecting style copying and helping artists argue their case (i.e. in a court of law) in such instances.

12.5 `ArtSavant`: A Practical Tool for Concerned Artists

We package DeepMatch and TagMatch into `ArtSavant`, a practical tool designed with a concerned artist in mind. Given a set of works by the concerned artist, `ArtSavant` would

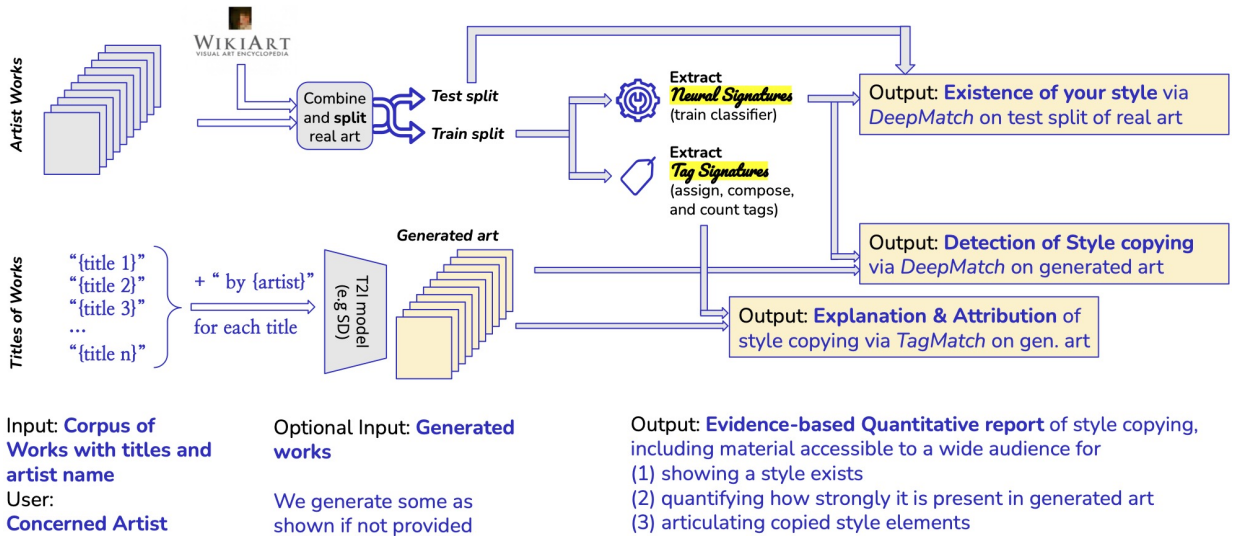


Figure 12.7: `ArtSavant` flow. We design our tool with a concerned artist in mind, who wishes to quickly investigate the degree to which they may be at risk of style copying by generative models.

create an easy-to-understand report characterizing the degree to which generative models copy the styles of the artist. As shown in Figure 12.7, the artist can present a set of generated images, or we can generate them by prompting text-to-image models with prompts of the form "{title of work} by {name of artist}". The provided works are then combined with our existing art repository and split into train/test sets. Using the train split, we (a) train a classifier over the $372 + 1$ artists, and (b) tag all images, compose tags within artists, and store extracted tag compositions per artist, resulting in neural and tag signatures. With these, we can apply DeepMatch and TagMatch respectively. Applying DeepMatch to the held-out art provides a measure of recognizability, establishing that the artist has an identifiable style to begin with. Then, running DeepMatch on generated images provides a quantitative manner to understand if (and to what degree) the artist's style appears consistently in generated works. Finally, running TagMatch on the generated images helps articulate the particular style signatures that are copied, enabling an analytic way to argue infringement, while also surfacing stylistically similar examples.

Figure 12.1 shows an example report outputted by `ArtSavant` when presented with art from an artist named Canaletto, who we observed was at risk of style infringement. We design the report to be easy to understand, quantitative, and legally-grounded. Moreover, the report can be generated very quickly: as all steps operate on embeddings from a frozen CLIP encoder, the process takes about 1-2 minutes.

12.5.1 Analysis with `ArtSavant`: Quantifying Style Copying of 372 Prolific Artists

While enough anecdotal instances of style mimicry have been observed to raise concern [340, 344], the prevalence and nature of such instances remains nebulous. To shed quantitative insight on style copying, we now leverage `ArtSavant`on the 372 artists from our WikiArt dataset, generating images with three popular text-to-image models: (i) Stable-Diffusion-v1.4; (ii) Stable-Diffusion-v2.0; and (iii) OpenJourney from PromptHero.

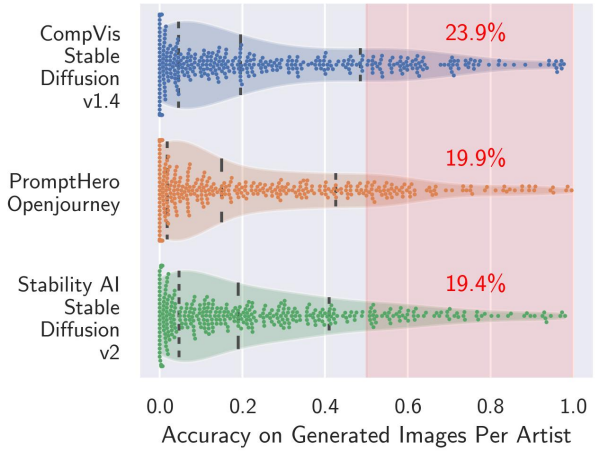


Figure 12.8: DeepMatch on generated art. In red: the fraction of artists with their styles recognized in at least half of their respective generated images.

Following figure 12.7, we employ a simple prompting strategy of augmenting painting titles with the name of the artist.

We first apply **DeepMatch** to see what fraction of artists' styles can be recognized consistently over generated images. Namely, each generated image is classified to one of 372 artists,

and per artist, predictions are aggregated via majority voting. Figure 12.8 shows the 'accuracy' on generated images per artist, where accuracy is now interpreted as the rate which images generated to copy an artist are classified as that artist. In red, the fraction of artists who see accuracies of at least 50% (i.e. so that the generated image *set* is classified to the original artist) are denoted per model, which we call the match rate. We observe an average match rate of 20.2%, indicating that for the vast majority of artists in our study, *simple prompting of generative models does not reproduce their styles* in a way recognizable to DeepMatch, which has an 89% match rate on real art. For all three models, over half the artists see accuracies below 20%, with 26% of artists seeing an average accuracy below 5% for generated images. On the other hand, a handful of artists' styles are matched with high confidence: 16 artists see average accuracies over 75%. These include ultra famous artists like Van Gogh, Claude Monet, Renoir, which we'd expect generative models to do well in emulating. However, a few relatively lesser known artists are also present, like Jacek Yerka, who are still alive, and thus could be negatively affected by generative models reproducing their styles.

We additionally design a small scale human evaluation to validate the outputs of our automated approach (see Appendix 12.7.3). We observe that generated art can fail to meet the threshold of style copying in two ways: either the generated art does not match the artist's style, or the style of the generated art is no more similar to the artist than to that of a highly similar artist. This underscores that style similarity alone does not constitute infringement, which instead requires replication of unique style signatures.

With **TagMatch**, in addition to predicting an artistic style, we can also articulate the specific tag signature shared between the test set of images and the reference set of images for the predicted style. Thus, we can inspect the shared signature, as well as instances from both sets

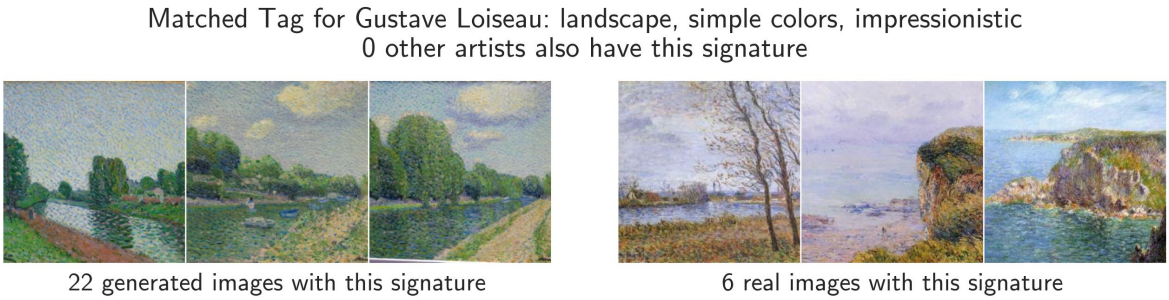


Figure 12.9: Examples of applying TagMatch to generated images. TagMatch is inherently interpretable with respect to tags, as each inference comes with the exact set of tags that are (i) shared between the sets of test art and art from the predicted artist, and (ii) used to predict the artist.

where the signature is present, providing direct evidence of the potential style infringement a broader audience to independently verify. Inspecting some examples in figure 12.9, we observe that while pixel level differences are common across retrieved image subsets, stylistic elements are consistent in both sets with the labeled tags, echoing our motivating claim that style copying goes beyond image or pixel-wise similarity. Lastly, TagMatch also allows for understanding image distributions from the perspective of interpretable tags.

12.6 Conclusion

In our paper, we rethink the problem of copyright infringement in the context of artistic styles, which currently have no legal protection. So that our analysis is valuable to the relevant stakeholders, we base our method in legal precedent and take steps to ensure the interpretability of our approach. Then, we study the scientific basis for three questions: **i.** Do unique artistic styles exist? **ii.** Can unique styles be articulated interpretably in a quantitatively-grounded manner? **iii.** Do these unique styles re-appear in generated art? We answer these questions empirically, using a simple logical argument at the heart of our methods: if an artist's works are consistently recognized as their own, the artist has a unique style. Having reformulating the task to a classification problem over image sets, we develop `ArtSavant` – a novel tool to extract and detect artistic style *signatures* in a manner that is quantitative, intepretable, and rooted in legal principles. Crucially, we develop `ArtSavant` not to replace human decision makers, but instead to aide them in navigating the nuance associated with style copying. We find evidence of the existence of artistic styles, and in an empirical study, quantify the degree to which styles are potentially infringed, validating our framework. We hope our contributions help bridge the gap

between legal and technical communities, towards quantitatively examining the nuanced issue of artistic style infringements.

12.7 Appendix

12.7.1 A Brief History of Copyright Law: Situating our work

Throughout its history, copyright law has been centrally concerned with and affected by the replacement of slow, expensive, inexact human copying with comparatively fast, cheap, and exact machine copying. The very origin of copyright law is tied to the replacement of hand copying of manuscripts by scribes to machine duplication with printing presses. As the use of presses grew, publishers and authors became aware that printing enabled wider distribution of books. At the same time, however, they became aware that inexpensive unauthorized copying of those books would threaten the compensation available to an author for their initial investment in writing a book [361, 362]. The first copyright laws – the Statute of Anne in England in 1710 [363], and the Copyright Act of 1790 [364] in the United States – grew directly out of those concerns.

As machine copying has spread to other types of creative works, copyright law has adapted by including those works as copyrightable subject matter. In the mid-nineteenth century, chromolithography enabled dramatically better and less expensive copying of paintings. Congress responded in 1870 by extending copyright protection to paintings [365, 366]. In the twentieth century, the development of audio and visual recording technologies presented new opportunities but also new threats to performers – musicians, actors, dancers, and other performing artists. Because recorded performances could in many cases be an adequate substitute for live performances, new markets opened, but both new and old markets would be threatened if recordings

could be made and distributed without authorization. Congress responded by extending federal copyright protection to motion pictures in 1912 [367], and to sound recordings in 1971 [368]. In the case of sound recordings, Congress also distinguished between human and machine copying. Federal copyright law permits human copying of the sounds in a sound recording, through human imitation of those sounds that are independently recorded, while it prohibits direct electronic or mechanical machine copying of a sound recording [369]. That is another layer of recognition of the particular threat to authors posed by inexpensive, exact machine copying.

Generative AI is the newest technology for making creative works. Current generative AI tools are not designed to enable copying of specified preexisting works. However, for the first time, they may replace expensive human imitation of individual artistic styles with inexpensive machine mimicking of those styles. There have always been talented human artists who, with enough time for study and painstaking manual imitation, can produce believable forgeries – works that are not copies of any previously known works of well-known artists, but that convincingly apply the style of those artists to a different subject matter. In the twentieth century, artists such as Han van Meegeren [370], Wolfgang and Helene Beltracchi [371], and Shaun Greenhalgh [372] became infamous for their ability to create convincing forgeries of works of other artists. However, those forgers took weeks or months to create their forgeries, and they were motivated to make that considerable investment of time only because they believed they could sell the results for tens or hundreds of thousands of dollars.

By contrast, generative AI tools can create images, and texts and music in minutes or hours, at total machine and labor costs of only tens or hundreds of dollars, rather than tens or hundreds of thousands of dollars. Generative AI training algorithms are capable of modeling extraordinarily complicated patterns in a set of training works, and thus there is some reason to

think that they could model the style of an author as manifested in a set of that author's works. There is also anecdotal evidence that generative AI tools can in fact produce new works that audiences mistakenly believe were created by particular human artists, authors, or performers [373, 374]. Those capabilities raise the issue of whether copyright law should step in and protect copying of individual artistic styles, as it has stepped in many times in the past to protect copying of different types of individual works. That could be accomplished through legislation, but also possibly judicially. Because copyright infringement standards have never been codified, courts have created and modified those standards themselves. In the case of protection of fictional characters, courts have already gone beyond the individual work of authorship, and protected structures or entities that are only fully developed in multiple works [334, 375, 376].

Our purpose in this paper is not to take a position on the issue of whether copyright law should be extended to protect individual artistic styles, or on whether such an extension could be accomplished through legislation or through judicial decision. Neither is our purpose to automate the analysis of copyright infringement. Rather, we are interested in investigating whether there is any scientific support for the idea that there are identifiable individual artistic styles, and that those styles could be correlated with a group of human-understandable stylistic terms. Anecdotes about a generative AI tool producing an image or text that some people come to believe was created by a particular human artist or author are not proof that all or even some artists or authors have stable individual styles that can be modelled and applied in a wide variety of images, text, or music. However, when we try to train a model that can correctly identify authors of previously unseen works, we may get closer to understanding whether and when individual artistic styles exist. If we can link that classification to a selection of terms describing characteristics of those works, then we can explain to human beings what the components of such individual artistic

styles might be. If some kind of protection of individual style ever became part of copyright law, judges or juries would still have to decide whether a particular output of a generative AI tool too closely mimicked the individual style of an artist. The degree to which an AI model could or could not correctly identify the author of a work, and the stylistic terms that that model could or could not correlate with that classification, would simply provide additional information to those human decision-makers.

12.7.2 A nuance in artistic style infringements: Existing Artists can have very similar styles

A crucial step in arguing that an artist's style has been infringed is to first demonstrate the existence of the given artist's *unique* style. We note that doing so objectively is non-trivial, as a style may not have a clear definition, and thus, it can be challenging to systematically compare to all other artistic styles, so to show uniqueness. In our work, we utilized classification, claiming that if an artist's works can consistently be mapped (i.e. at least half the time) to that artist (over a large set of other artists), than that artist must have some underlying unique style (parameterized by a neural signature).

In doing so, we found that 89.3% of artists could be recognized based of a set of (at least 20 of) their works (held-out in training the classifier). What about the remaining 10.7% of artists? We now take a closer look at these artists, and also introduce a second, stricter style copying criterion. Namely, we consider the notion that it may be unfair to claim a generative model is copying the style of an artist, if another existing artist seems to also be copying that artist. That is, we propose a way to verify that the generative model not only shows a substantial similarity

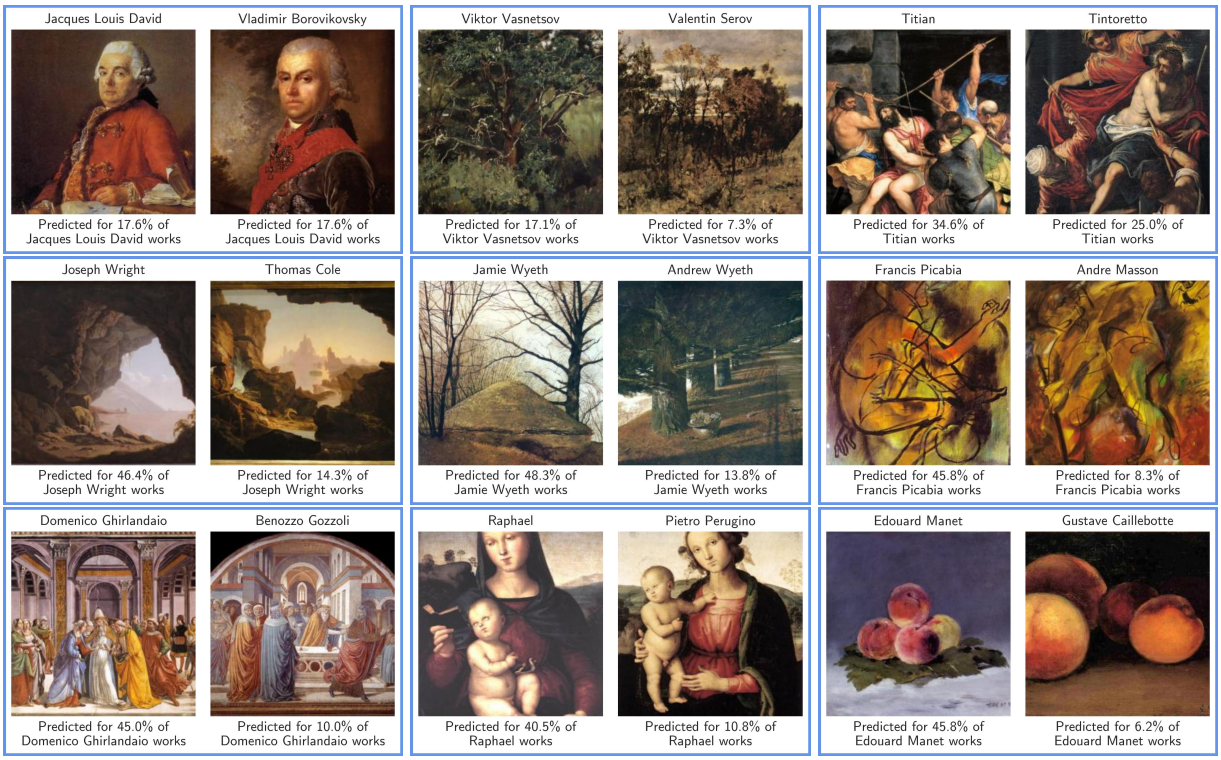


Figure 12.10: Examples of artists who's styles were not recognized by DeepMatch (i.e. less than half of their held-out works were predicted to the artist). Each panel shows an example work from (left) the unrecognized artist and (right) the artist that is incorrectly predicted most frequently over works from the unrecognized artist. We see that artists can use very similar, at times arguably indistinguishable, styles.

to the copied artist, but also an *unprecedented* similarity.

12.7.2.1 Artists who's styles were not recognized

First, we inspect more examples from artists who were not recognized using our majority voting threshold in DeepMatch. That is, less than half of their held-out works were predicted to them. Figure 12.10 shows a number of examples, from which we can make some qualitative observations. First, the styles of artists who operate in the same broader genre (e.g. portraiture, landscapes, narrative scenes in renaissance styles, etc) can be extremely similar. We even see an instance where an artist's son's style is indistinguishable from his father's (Jamie and Andrew

Wyeth). Lastly, we note that in most cases, the artists only marginally fall short of our recognition threshold (i.e. accuracy for their held-out works is only a bit below 50%). We utilize majority voting because (i) it is intuitive, (ii) it requires *consistent* appearance of the neural signature across works, and (iii) it allows for abstention when no particular style is strongly present. However, the exact threshold of 50% can be altered as desired. In summary, as in Figure 12.4, we see artistic styles can be very similar, making the existence of unique artistic styles for the vast majority of artists a non-trivial observation.

If an artist's style cannot be recognized over their own held-out works, arguing that a generative model copies that style is strenuous, as the style itself is ill-defined. Notably, in these cases, the classifier had an option to predict the correct artist. However, in applying DeepMatch to generated images, there is no direct option for the classifier to abstain from predicting anyone, under that generated art comes from a "new artist", which takes inspiration from existing artists. Note that abstention is still possible (due to the majority voting in DeepMatch), and occurs when a match confidence falls below 50%. To make comparisons fairer to generative models, we now discuss a stricter criterion of *unprecedented similarity*.

12.7.2.2 *Unprecedented Similarity*: Do generative models copy styles more than existing artists already do?

A nuance that requires consideration when studying artistic style copying is that it is possible for two artists to have very similar styles. Thus, it may be unfair to allege that a generative model is copying an artist a if there exists another artist b who's style is just as or in fact even more similar to artist a. Towards this end, we introduce *unprecedented similarity*, which requires that

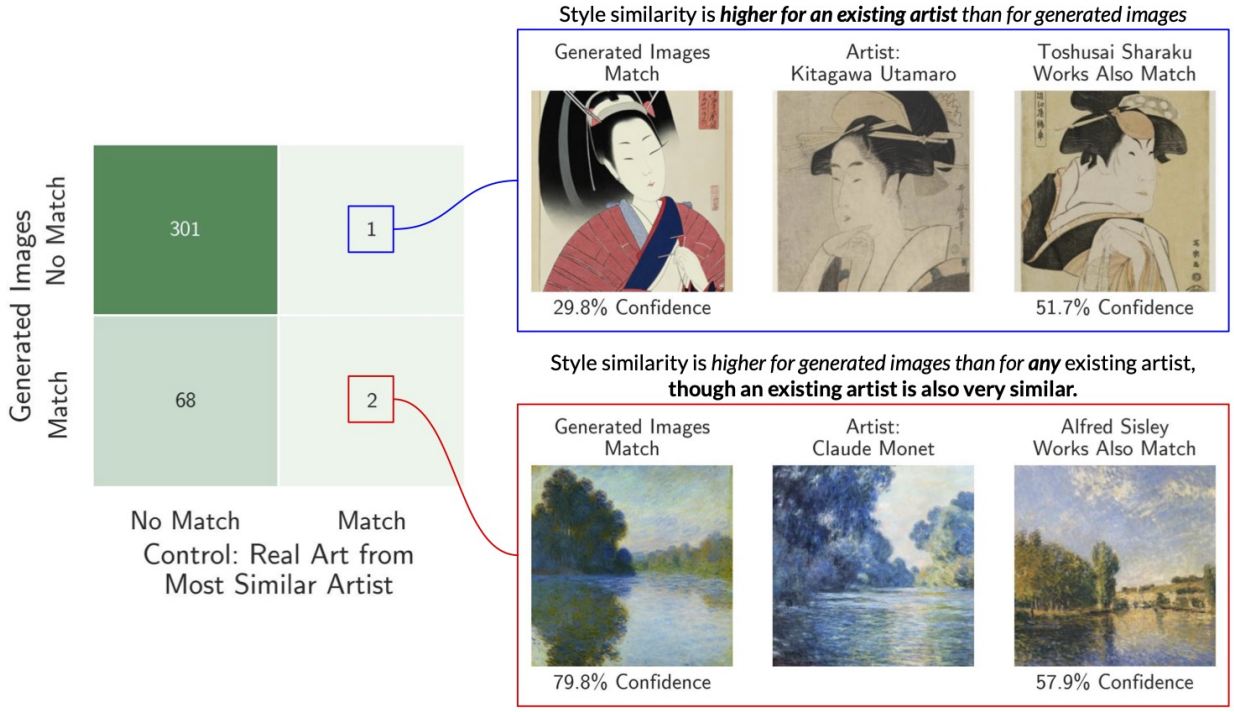


Figure 12.11: We verify the stricter criterion of *unprecedented similarity* by holding out the real artist with highest similarity to a given artist, and checking if the held-out real artist's works are flagged as potential style copying by DeepMatch. (**left**) We observe only three artists where the most similar held-out artist has their work flagged as a style match, and in all cases, when generated images are flagged, the match confidence of the generated images exceeds that of the held-out real artist's works (i.e., **the generated images flagged by our method reflect *unprecedented similarity* to the given artist's style**). (**right**) Inspecting the flagged held-out artists further show that style copying is very nuanced, as artists take inspiration from one another, and as such, they may already have very similar styles. While we always observe unprecedented similarity, a potential solution to style copying may be for generative models to ensure that they do not copy any more than what already exists; that is, they may exhibit some copying, but no more than for which precedent already exists.

the similarity between works of a generative model A' and works of the artist inteded to be copied A is higher than the similarity of any existing artist with A. That is, $sim(A, A') \geq sim(A, B)$ for works B from all other existing artists b.

Note that this is a stricter criterion than our previous threshold. In DeepMatch, we required that at least half of the works in a given set of test images were predicted to a single artist in order for us to flag the test images as a potential style infringmenet. In other words, that threshold

required that $sim(A, A') \geq 0.5$, which in turn implies that $sim(A, A') \geq sim(A', B)$ for all B (with room to spare; here we use match confidence to denote similarity).

Now, however, instead of just comparing A' to all B, we must also compare all B to A. Instead of comparing all other artists, we inspect the most similar artist b^* to a, identified by taking the artist b with the highest rate of false positive predictions to artist a. Then, we hold out b, and train a new classifier on the remaining 371 artists. Finally, we check for style matches of for the set of generated images A' and the works B^* from the most similar artist b^*.

Figure 12.11 summarizes our result for OpenJourney (all three models studied show consistent results). We find that only in three cases do we see a held-out artist's work flagged as potential style copying. Notably, in all instances where generated work is flagged as potential style copying, the corresponding held-out artist's work is either not flagged or is flagged with lower confidence, indicating that the instances of style copying of generative models that we observe always also satisfy the criterion of unprecedented similarity.

Taking a closer look at instances where held-out art is flagged for style copying (or perhaps style emulation?), we again see just how similar the works of different artists can be. Namely, we see that some artists works seem to fall into a broader genre of art that many artists utilize (e.g. ukiyo-e or impressionism). In summary, while generative models can very closely resemble the style of a given artist, contextualizing copying by generative models with respect to copying (or perhaps, 'style emulation') already done by existing artists is crucial in order to afford the same artistic liberties to generative models as have been provided to other artists in the past.

12.7.3 Human Validation of Style Copying

We design and conduct a human study to assess style copying, toward verifying the outputs of our system. Given a 'plaintiff' artist A, we present a set of artworks generated by Stable Diffusion v1.4 in the style of the plaintiff A, along with two sets of real artworks: one from the plaintiff A, and another from a very similar artist B (selected as the artist – aside from A – that DeepMatch predicts to have created the highest number of the artworks generated to be in the style of A).

A human then denotes if the set of generated art more closely resembles the style of set from artist A or artist B, with the choice to abstain if neither set is more similar to the generated art than the other. Abstention may occur if the generated art is different from both sets or similar to both sets. Each set contains 16 works. The 16 generated artworks are selected at random from the collection of images generated in artist A's style. The 16 images each from artist A and B are drawn to be those that are most similar (via mean CLIP similarity) to the 16 selected generated images. The two sets of real art are presented in random order so that the human does not know which belongs to the plaintiff artist.

We run this experiment for 40 artists: 20 which are flagged by our method as being at risk of style copying, and 20 which are not – we call the first group SAFE and the second group COPIED. We find that **the percent of artists where the generated art is marked as more similar to the plaintiff is 90% for the COPIED group and just 5% for the SAFE group**. Notably, the human abstained in 19 of the 20 artists in the SAFE group: for 15 artists, the generated art was dissimilar to both sets of real art, while for 4 artists, the generated art was equally similar to both sets. The latter type of abstentions underscores an important point: generated art can be similar to a plaintiff

artist without necessarily meeting a bar for infringement, since infringement would require that a unique style is copied. This point relates to the discussion of CSD: while an improved style similarity metric can help surface relevant art, on its own, it does not immediately lead to an understanding of if an artist's style is unique, and what elements comprise that style. These questions are critical when it comes to copyright protection for artistic styles, and we center answering them in our work.

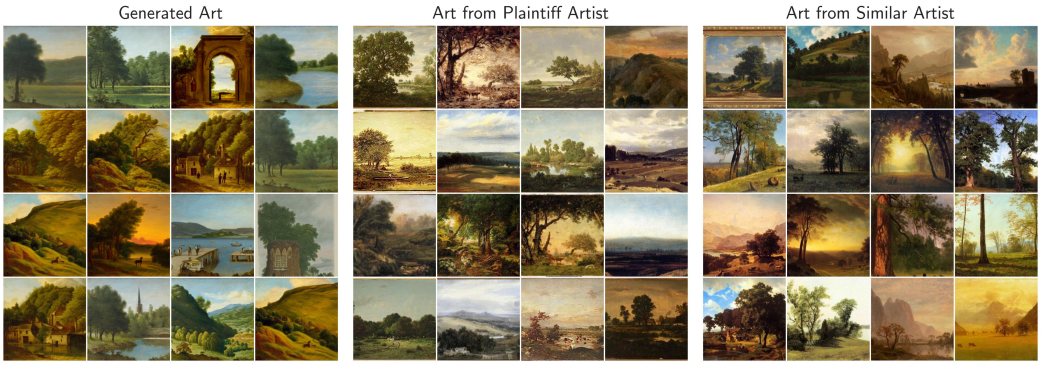
In summary, our human study corroborates the outputs of our method. Artists that are flagged to be at risk of style copying have generated artwork that is more similar to their work than even art from the most similar other artist. However, artists who are not flagged by our method have generated artwork that is either dissimilar to their style, or no more similar to their style than to that of another artist.



(a) Case 1: Generated art from *Pierre Auguste Renoir* was deemed more similar to art by Renoir than art from a similar artist. Here, the generated art seems to replicate aspects of Renoir's style.



(b) Case 2: Generated art from *Sally Gabori* was deemed dissimilar to both real art from Gabori and another artist. Here, the generated art does not seem to be aware of Gabori's style.



(c) Case 3: Generated art from *Theodore Rousseau* was deemed similar to both real art from Rousseau and from another artist, leading the human annotator to abstain from saying that the generated art copies Rousseau in particular. This highlights how style similarity may not constitute infringement.

Figure 12.12: Samples visualizations from our human evaluation. In the real evaluation, panel titles are removed and order is randomized. The first case demonstrates an instance where the human annotator viewed the generated art as more stylistically similar to the plaintiff's art than the most similar other artist, suggesting that Renoir's unique style is at risk of copying. The latter two cases show different ways in which the threshold of style copying may not be met: when the generated art is not similar to the plaintiff, or when the generated art is not sufficiently *uniquely* similar to the plaintiff.

Chapter 13: Looking ahead

The work to ensure the reliability of AI never ends. The past four years have seen a dramatic rise in the visibility, capabilities, and accessibility of AI models. Soon, many new applications will be proposed, embedding AI into the daily lives of more people than ever before. From this dissertation, I hope a few lessons can be taken moving forward. First, the limitations of AI models can be surprising. Second, the space of potential failure modes is vast – so much so that exclusively relying on manual hypothesis and verification of limitations would be ill-advised. Fortunately, machines are adept at handling scale, and so, while it may feel circular to use models to audit other models, the potential for human-AI collaboration in tackling the intricate and formidable task of ensuring AI reliability should not be overlooked. Finally – and perhaps most importantly – **science does not exist in a vacuum**. AI has incredible potential to positively impact everyday people. However, this impact does not happen on its own, and unintended or disparate impacts can occur if we are not careful. So, I hope all scientists, especially those in AI, advance with consideration for how their work fits in the broader context of the society around us. If we wish for AI to help all of humanity, we must recognize that people's needs, constraints, and opportunities vary widely — and build with that reality in mind. Let us then engage folks from all communities and professions, understand the problems that need solutions (not the other way around), and incorporate their unique insights, so that we can truly build a technology that lifts all of us up.

Bibliography

[1] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XVPqLyNxSyh.

[2] William A Gaviria Rojas, Sudnya Diamos, Keertan Ranjan Kini, David Kanter, Vijay Janapa Reddi, and Cody Coleman. The dollar street dataset: Images representing the geographic and socioeconomic diversity of the world. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=qnfYsave0U4.

[3] Alec Tyson and Emma Kimuchi. Growing public concern about the role of artificial intelligence in daily life. *Pew Research Center*, 2023. URL https://www.pewresearch.org/short-reads/2023/08/28/growing-\public-concern-about-the-role-of-artificial-intelligence\-in-daily-life/.

[4] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

[5] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.

[6] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk — quantifying and controlling the effects of context in classification and segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8210–8218, 2019.

[7] Amir Rosenfeld, Richard S. Zemel, and John K. Tsotsos. The elephant in the room. *ArXiv*, abs/1808.03305, 2018.

[8] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38, 2022. URL https://api.semanticscholar.org/CorpusID:246652372.

[9] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[10] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Koulako Bala Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel J. Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258, 2021. URL `https://api.semanticscholar.org/CorpusID:237091588`.

[11] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[14] J. Deng, Wei Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[15] Mazda Moayeri, Phillip E. Pope, Yogesh Balaji, and Soheil Feizi. A comprehensive study of image classification model sensitivity to foregrounds, backgrounds, and visual attributes. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19065–19075, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[18] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[19] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[22] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

[23] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015. URL http://arxiv.org/abs/1512.04150.

[24] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

[25] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

[26] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

[27] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.

[28] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.

[30] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[31] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010.

[32] Wanli Ouyang, Hongyang Li, Xingyu Zeng, and Xiaogang Wang. Learning deep representation with large-scale attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[34] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028, 2021.

[35] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[36] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[37] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019.

[38] Jessica Yung, Rob Romijnders, Alexander Kolesnikov, Lucas Beyer, Josip Djolonga, Neil Houlsby, Sylvain Gelly, Mario Lucic, and Xiaohua Zhai. Si-score: An image dataset for fine-grained analysis of robustness to object location, rotation and size. *CoRR*, abs/2104.04191, 2021. URL `https://arxiv.org/abs/2104.04191`.

[39] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=gl3D-xY7wLq`.

[40] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ArXiv*, abs/1911.08731, 2019.

[41] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1885–1894. JMLR.org, 2017.

[42] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11205–11216. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wong21b.html.

[43] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

[44] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688, Jul. 2019. doi: 10.1609/aaai.v33i01.33013681. URL https://ojs.aaai.org/index.php/AAAI/article/view/4252.

[45] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *ArXiv*, abs/2006.14651, 2021.

[46] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

[47] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *CoRR*, abs/1602.03616, 2016. URL http://arxiv.org/abs/1602.03616.

[48] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12853–12862, 2021.

[49] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

[50] Chun-Hao Chang, George Alexandru Adam, and Anna Goldenberg. Towards robust classification model by counterfactual and invariant data generation. *arXiv preprint arXiv:2106.01127*, 2021.

[51] Andreas Geiger Axel Sauer. Counterfactual generative networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[52] Sven Gowal, Chongli Qin, Po-Sen Huang, A. Taylan Cemgil, Krishnamurthy Dvijotham, Timothy A. Mann, and Pushmeet Kohli. Achieving robustness in the wild via adversarial mixing with disentangled representations. *CoRR*, abs/1912.03192, 2019. URL http://arxiv.org/abs/1912.03192.

[53] Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[54] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *CoRR*, abs/2106.00545, 2021. URL https://arxiv.org/abs/2106.00545.

[55] Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge J. Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. *CoRR*, abs/2004.12276, 2020. URL https://arxiv.org/abs/2004.12276.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. URL http://arxiv.org/abs/1405.0312. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

[57] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[58] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[59] Tanushree Mitra, C.J. Hutto, and Eric Gilbert. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1345–1354, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702553. URL https://doi.org/10.1145/2702123.2702553.

[60] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. URL https://arxiv.org/abs/2012.12877.

[62] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *CoRR*, abs/2103.15670, 2021. URL https://arxiv.org/abs/2103.15670.

[63] Philipp Benz, Soomin Ham, Chaoning Zhang, Adil Karjauv, and In So Kweon. Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. *arXiv preprint arXiv:2110.02797*, 2021.

[64] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *CoRR*, abs/2103.14586, 2021. URL https://arxiv.org/abs/2103.14586.

[65] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners, 2021.

[66] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers, 2021. URL https://arxiv.org/abs/2105.10497.

[67] Tianlu Wang, Diyi Yang, and Xuezhi Wang. Identifying and mitigating spurious correlations for improving robustness in nlp models, 2021.

[68] Mazda Moayeri, Sahil Singla, and Soheil Feizi. Hard imagenet: Segmentations for objects with strong spurious cues. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=76w7bsdViZf.

[69] Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

[70] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony Beardsworth Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15, 2018.

[71] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019.

[72] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

[73] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Sara Beery, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv:2012.07421*, 2020.

[74] Sahil Singla, Mazda Moayeri, and Soheil Feizi. Core risk minimization using salient imagenet, 2022. URL https://arxiv.org/abs/2203.15566.

[75] Adam Coates, A. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[76] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *ArXiv*, abs/2204.02937, 2022.

[77] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *ArXiv*, abs/2006.07710, 2020.

[78] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

[79] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, 2021.

[80] Alexander Robey, George J. Pappas, and Hamed Hassani. Model-based domain generalization. In *NeurIPS*, 2021.

[81] Mengye Ren, Wenyuan Zeng, Binh Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[82] J. Peters, Peter Buhlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78, 2015.

[83] Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. 2021.

[84] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2021.

[85] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2018. URL `https://arxiv.org/abs/1811.12231`.

[86] Yue He, Zheyan Shen, and Peng Cui. Towards non-iid image classification: A dataset and baselines. *Pattern Recognition*, 110:107383, 2021.

[87] Priyatham Kattakinda and Soheil Feizi. Focus: Familiar objects in common and uncommon settings. *ArXiv*, abs/2110.03804, 2021.

[88] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2018.

[89] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2009.

[90] Judy Borowski, Roland S. Zimmermann, Judith Schepers, Robert Geirhos, Thomas S. A. Wallis, Matthias Bethge, and Wieland Brendel. Exemplary natural images explain cnn activations better than feature visualizations. *ArXiv*, abs/2010.12606, 2020.

[91] Mazda Moayeri, Kiarash Banihashem, and Soheil Feizi. Explicit tradeoffs between adversarial and natural distributional robustness, 2022. URL https://arxiv.org/abs/2209.07592.

[92] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony Beardsworth Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15, 2018.

[93] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

[94] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dFwBosAcJkN.

[95] Hongyang R. Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[96] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya P. Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *ArXiv*, abs/1906.04584, 2019.

[97] Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. *ArXiv*, abs/2006.00731, 2020.

[98] Alexander Levine and Soheil Feizi. Improved, deterministic smoothing for l1 certified robustness. In *ICML*, 2021.

[99] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[100] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *CoRR*, abs/2001.03994, 2020. URL https://arxiv.org/abs/2001.03994.

[101] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

[102] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Scholkopf, and Kun Zhung. Causaladv: Adversarial robustness through the lens of causality. In *ICLR*, 2022.

[103] Cynthia H Zhang, Kun Zhang, and Yingzhen Li. A causal view on robustness of neural networks. *ArXiv*, abs/2005.01095, 2020.

[104] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

[105] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *ArXiv preprint arXiv:2007.08489*, 2020.

[106] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[107] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.

[108] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[109] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *ICML*, 2019.

[110] Nicolas Papernot, Patrick Mcdaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2016.

[111] Mazda Moayeri and Soheil Feizi. Sample efficient detection and classification of adversarial attacks via self-supervised embeddings. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7657–7666, 2021.

[112] Wei-An Lin, Chun Pong Lau, Alexander Levine, Ramalingam Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. *ArXiv*, abs/2009.02470, 2020.

[113] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[114] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

[115] David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021.

[116] Alexander Kolesnikov and Christoph H. Lampert. Improving weakly-supervised object localization by micro-annotation. *ArXiv*, abs/1605.05538, 2016.

[117] Michael A. Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh M Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4849, 2019.

[118] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan I. Moldovan, Sylvan Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16453–16463, 2021.

[119] Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML*, 2018.

[120] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5 2:153–163, 2017.

[121] J. Zhang, Aditya Krishna Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Suvrit Sra. Coping with label shift via distributionally robust optimisation. *ArXiv*, abs/2010.12230, 2021.

[122] Benjamin Aubin, Agnieszka Slowik, Martín Arjovsky, Léon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *ArXiv*, abs/2102.10867, 2021.

[123] E. Dobriban, Hamed Hassani, David Hong, and Alexander Robey. Provable tradeoffs in adversarially robust classification. *ArXiv*, abs/2006.05161, 2020.

[124] Alireza Mousavi Hosseini, Amir Mohammad Abouei, and Mohammad Hossein Rohban. The interplay between distribution parameters and the accuracy-robustness tradeoff in classification. *arXiv preprint arXiv:2107.00247*, 2021.

[125] Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pages 2034–2078. PMLR, 2020.

[126] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019.

[127] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[128] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses, 2020.

[129] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

[130] Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

[131] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020. doi: 10.1109/ACCESS.2020.3010274.

[132] Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Perturbation type categorization for multiple ℓ_p bounded adversarial robustness, 2021. URL https://openreview.net/forum?id=Oe2XI-Aft-k.

[133] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[134] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017.

[135] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

[136] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability, 2021.

[137] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7614–7623, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/zhu19a.html.

[138] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21945–21957. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f9fd2624beefbc7808e4e405d73f57ab-Paper.pdf.

[139] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1702.04267.

[140] Fabio Carrara, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato. Adversarial examples detection in features distance spaces. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 313–327, Cham, 2019. Springer International Publishing. ISBN 978-3-030-11012-3.

[141] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples, 2017.

[142] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/roth19a.html.

[143] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1gJ1L2aW.

[144] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017. URL http://arxiv.org/abs/1704.01155.

[145] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. 10 2017.

[146] D. J. Miller, Y. Wang, and G. Kesidis. Anomaly detection of attacks (ada) on dnn classifiers at test time. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. doi: 10.1109/MLSP.2018.8517069.

[147] Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, D. Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *ArXiv*, abs/2012.01654, 2020.

[148] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners, 2020.

[149] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020.

[150] Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=bgQek2O63w.

[151] Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17081–17093. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c68c9c8258ea7d85472dd6fd0015f047-Paper.pdf.

[152] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2983–2994. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1f1baa5b8edac74eb4eaa329f14a0361-Paper.pdf.

[153] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16199–16210. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/ba7e36c43aff315c00ec2b8625e3b719-Paper.pdf.

[154] Vikash Sehwag, Mung Chiang, and Prateek Mittal. {SSD}: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=v5gjXpmR8J.

[155] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[156] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks, 2020.

[157] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment, 2023.

[158] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[159] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17897–17907, 2021.

[160] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[161] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021.

[162] Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

[163] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[164] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/77d2afcb31f6493e350fca61764efb9a-Paper.pdf.

[165] Thomas Fel, Agustin Picard, Louis Béthune, Thibaut Boissin, David Vigouroux, Julien Colin, R'emi Cadene, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. *ArXiv*, abs/2211.10154, 2022.

[166] Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A. Ehinger, and Benjamin I. P. Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *AAAI Conference on Artificial Intelligence*, 2020.

[167] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022.

[168] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space. *arXiv preprint arXiv:2206.14754*, 2022.

[169] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18102–18112, 2021.

[170] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. *arXiv preprint arXiv:2209.15162*, 2022.

[171] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. *ArXiv*, abs/2209.15430, 2022.

[172] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv: Computer Vision and Pattern Recognition*, 2020. URL https://api.semanticscholar.org/CorpusID:221095529.

[173] Constantin Eichenberg, Sid Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. Magma - multimodal augmentation of generative models through adapter-based finetuning. *ArXiv*, abs/2112.05253, 2021.

[174] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734, 2021.

[175] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[176] Mazda Moayeri, Vidhisha Balachandran, Varun Chandrasekaran, Safoora Yousefi, Thomas Fel, Soheil Feizi, Besmira Nushi, Neel Joshi, and Vibhav Vineet. Unearthing skill-level insights for understanding trade-offs of foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kNHVViEPWK.

[177] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[178] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642, 2015.

[179] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fibxvahvs3.

[180] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

[181] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

[182] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024. Accessed: 2024-09-18.

[183] Gemini-Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

[184] Anthropic. Introducing claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, 2024. Accessed: 2024-09-18.

[185] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi, 2024.

[186] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023.

[187] Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, 2020.

[188] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[189] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings, 2022.

[190] Keivan Rezaei, Mehrdad Saberi, Mazda Moayeri, and Soheil Feizi. PRIME: Prioritizing interpretability in failure mode extraction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QrEHs9w5UF.

[191] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

[192] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213, 2022.

[193] Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *ArXiv*, abs/2402.01761, 2024. URL https://api.semanticscholar.org/CorpusID:267412530.

[194] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O. Riedl. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 263–274. Association for Computing Machinery, 2019. doi: 10.1145/3301275.3302316.

[195] Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason, 2023.

[196] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

[197] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

[198] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KRLUvxh8uaX.

[199] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3170–3180, October 2023.

[200] Mazda Moayeri, Michael Rabbat, Mark Ibrahim, and Diane Bouchacourt. Embracing diversity: Interpretable zero-shot classification beyond one vector per class. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, page 2302–2321. Association for Computing Machinery, 2024.

[201] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification.

[202] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. 2019.

[203] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón,

Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swedrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj.

[204] Haider Al-Tahan, Quentin Garrido, Randall Balestriero, Diane Bouchacourt, Caner Hazir-

bas, and Mark Ibrahim. Unibench: Visual reasoning requires rethinking vision-language beyond scaling, 2024. URL https://arxiv.org/abs/2408.04810.

[205] Vidhisha Balachandran, Jingya Chen, Neel Joshi, Besmira Nushi, Hamid Palangi, Eduardo Salinas, Vibhav Vineet, James Woffinden-Luey, and Safoora Yousefi. Eureka: Evaluating and understanding large foundation models, 2024. URL https://arxiv.org/abs/2409.10566.

[206] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-free llm benchmark, 2024. URL https://arxiv.org/abs/2406.19314.

[207] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[208] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International conference on machine learning*. PMLR, 2024.

[209] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

[210] Mazda Moayeri, Elham Tabassi, and Soheil Feizi. Worldbench: Quantifying geographic disparities in llm factual recall. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24. Association for Computing Machinery, 2024.

[211] Badr Youbi Idrissi, Diane Bouchacourt, Randall Balestriero, Ivan Evtimov, Caner Hazirbas, Nicolas Ballas, Pascal Vincent, Michal Drozdzal, David Lopez-Paz, and Mark Ibrahim. Imagenet-x: Understanding model mistakes with factor of variation annotations, 2022. URL https://arxiv.org/abs/2211.01866.

[212] Jieyu Zhang, Weikai Huang, Zixian Ma, Oscar Michel, Dong He, Tanmay Gupta, Wei-Chiu Ma, Ali Farhadi, Aniruddha Kembhavi, and Ranjay Krishna. Task me anything. *arXiv preprint arXiv:2406.11775*, 2024.

[213] Florian Bordes, Shashank Shekhar, Mark Ibrahim, Diane Bouchacourt, Pascal Vincent, and Ari S. Morcos. Pug: Photorealistic and semantically controllable synthetic data for representation learning, 2023.

[214] Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model, 2024. URL https://arxiv.org/abs/2405.02363.

[215] Vishvak Murahari, A. Deshpande, Peter Clark, Tanmay Rajpurohit, Ashish Sabharwal, Karthik Narasimhan, and A. Kalyan. Qualeval: Qualitative evaluation for model improvement. *ArXiv*, abs/2311.02807, 2023. URL https://api.semanticscholar.org/CorpusID:265033603.

[216] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1550–1553. IEEE, 2019.

[217] Sabri Eyuboglu, Karan Goel, Arjun Desai, Lingjiao Chen, Mathew Monfort, Chris Ré, and James Zou. Model changelists: Characterizing updates to ml models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24. Association for Computing Machinery, 2024.

[218] Eric Slyman, Minsuk Kahng, and Stefan Lee. Vlslice: Interactive vision-and-language slice discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15291–15301, October 2023.

[219] Besmira Nushi, Ece Kamar, and Eric Horvitz. Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6, pages 126–135, 2018.

[220] Lisa Dunlap, Yuhui Zhang, Xiaohan Wang, Ruiqi Zhong, Trevor Darrell, Jacob Steinhardt, Joseph E Gonzalez, and Serena Yeung-Levy. Describing differences in image sets with natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24199–24208, 2024.

[221] Richard T. White. Research into learning hierarchies. *Review of Educational Research*, 43(3):361–375, 1973. ISSN 00346543, 19351046. URL http://www.jstor.org/stable/1170110.

[222] Robert M. Gagne. The acquisition of knowledge. *Psychological Review*, 69(4):355–365, 1962. doi: 10.1037/H0042650.

[223] Kenneth R. Koedinger, Paulo F. Carvalho, Ran Liu, and Elizabeth A. McLaughlin. An astonishing regularity in student learning rate. *Proceedings of the National Academy of Sciences*, 120(13):e2221311120, 2023.

[224] Sanjeev Arora and Anirudh Goyal. A theory for emergence of complex skills in language models, 2023. URL https://arxiv.org/abs/2307.15936.

[225] Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. SKILL-MIX: a flexible and expandable family of evaluations for AI models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Jf5gplvglq.

[226] Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving, 2024. URL https://arxiv.org/abs/2405.12205.

[227] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

[228] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_3ELRdg2sgI.

[229] Satyapriya Krishna, Jiaqi Ma, Dylan Z Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. Post hoc explanations of language models can improve language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3H37XciUEv.

[230] Shuaibo Hu and Kui Yu. Learning robust rationales for model explainability: A guidance-based approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38, 2024. doi: 10.1609/aaai.v38i16.29783. URL https://ojs.aaai.org/index.php/AAAI/article/view/29783.

[231] Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H. Gilpin. Can large language models explain themselves? a study of llm-generated self-explanations, 2023. URL https://arxiv.org/abs/2310.11207.

[232] Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.19.

[233] Mohsen Fayyaz, Fan Yin, Jiao Sun, and Nanyun Peng. Evaluating human alignment and model faithfulness of llm rationale, 2024. URL https://arxiv.org/abs/2407.00219.

[234] Sam Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, John Kernion, Jamie Kerr, Jared Mueller, Jeff Ladish, Joshua D. Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noem'i Mercado, Nova Dassarma, Robin Larson, Sam McCandlish, Sandip Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom B. Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Benjamin Mann, and Jared Kaplan. Measuring progress on scalable oversight for large language models. *ArXiv*, abs/2211.03540, 2022. URL https://api.semanticscholar.org/CorpusID:253384413.

[235] Keivan Rezaei, Mehrdad Saberi, Mazda Moayeri, and Soheil Feizi. Prime: Prioritizing interpretability in failure mode extraction, 2023.

[236] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*, 2020.

[237] Vijay Vasudevan, Benjamin Caine, Raphael Gontijo Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. When does dough become a bagel? analyzing the remaining mistakes on imagenet. *Advances in Neural Information Processing Systems*, 35:6720–6734, 2022.

[238] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

[239] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[240] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[241] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[242] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3, 2022.

[243] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[244] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[245] Priyatham Kattakinda, Alexander Levine, and Soheil Feizi. Invariant learning via diffusion dreamed distribution shifts. *arXiv preprint arXiv:2211.10370*, 2022.

[246] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Bias-to-text: Debiasing unknown visual biases through language interpretation, 2023.

[247] Greg d'Eon, Jason d'Eon, James R. Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models, 2021.

[248] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011 (cub-200-2011). Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[249] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *ArXiv*, abs/2104.08718, 2021. URL `https://api.semanticscholar.org/CorpusID:233296711`.

[250] Mazda Moayeri, Wenxiao Wang, Sahil Singla, and Soheil Feizi. Spuriosity rankings: Sorting data to measure and mitigate biases. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=jSuhnO9QJv`.

[251] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *ArXiv*, abs/2206.07682, 2022.

[252] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[253] Chaofan Chen, Oscar Li, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. *ArXiv*, abs/1806.10574, 2019.

[254] Jonathan Crabbe, Zhaozhi Qian, Fergus Imrie, and Mihaela van der Schaar. Explaining latent representations with a corpus of examples. In *NeurIPS*, 2021.

[255] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind W. Picard. Dissect: Disentangled simultaneous explanations via concept traversals. *ArXiv*, abs/2105.15164, 2022.

[256] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *ArXiv*, abs/2202.00622, 2022.

[257] Jun Hyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. *ArXiv*, abs/2204.02070, 2022.

[258] Michael Zhang, Nimit Sharad Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher R'e. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. In *ICML*, 2022.

[259] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. *ArXiv*, abs/2201.11114, 2022.

[260] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11205–11216. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wong21b.html.

[261] Thomas Fel, Ivan Felipe, Drew Linsley, and Thomas Serre. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[262] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4352–4360. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.463. URL https://doi.org/10.1109/CVPR.2017.463.

[263] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258615488.

[264] John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. *ArXiv*, abs/2107.04649, 2021.

[265] Lucas Beyer, Olivier J. H'enaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *ArXiv*, abs/2006.07159, 2020.

[266] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2340–2350, 2021.

[267] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization, 2023.

[268] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2023.

[269] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-cqvvvb-NkI.

[270] OpenAI. Openai api, 2023. URL https://openai.com/product.

[271] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. March 15 2023. Available at SSRN: https://ssrn.com/abstract=4389233 or http://dx.doi.org/10.2139/ssrn.4389233.

[272] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems. *ArXiv*, abs/2303.13375, 2023. URL https://api.semanticscholar.org/CorpusID:257687695.

[273] Jungo Kasai, Yuhei Kasai, Keisuke Sakaguchi, Yutaro Yamada, and Dragomir Radev. Evaluating gpt-4 and chatgpt on japanese medical licensing examinations, 2023.

[274] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023.

[275] Terrance DeVries, Ishan Misra, Changhan Wang, and Laurens van der Maaten. Does object recognition work for everyone?, 2019.

[276] Laura Gustafson, Megan Richards, Melissa Hall, Caner Hazirbas, Diane Bouchacourt, and Mark Ibrahim. Pinpointing why object recognition performance degrades across income levels and geographies, 2023.

[277] Jessica Ojo, Kelechi Ogueji, Pontus Stenetorp, and David I. Adelani. How good are large language models on african languages?, 2023.

[278] World Bank. World bank open data, 2024. URL https://data.worldbank.org/.

[279] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. URL https://api.semanticscholar.org/CorpusID:259950998.

[280] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[281] Gemini Team. Gemini: A family of highly capable multimodal models, 2023.

[282] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023.

[283] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge, 2023.

[284] Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?, 2023.

[285] Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F. Rousseau, and Greg Durrett. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors, 2023.

[286] Yifan Zhang, Cheng Wei, Shangyou Wu, Zhengting He, and Wenhao Yu. Geogpt: Understanding and processing geospatial tasks through an autonomous gpt. *ArXiv*, abs/2307.07930, 2023. URL https://api.semanticscholar.org/CorpusID:259937048.

[287] Melissa Hall, Candace Ross, Adina Williams, Nicolas Carion, Michal Drozdzal, and Adriana Romero-Soriano. DIG in: Evaluating disparities in image generations with indicators for geographic diversity. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FDt2UGM1Nz. Featured Certification.

[288] World Bank. World bank country and lending groups, 2024. URL https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups#:~:text=For%20the%20current%202024%20fiscal,those%20with%20a%20GNI%20per.

[289] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[290] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[291] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

[292] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[293] Cohere AI. Cohere api, 2023. URL `https://github.com/cohere-ai/cohere-python`.

[294] Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Tauman Kalai. Do language models know when they're hallucinating references?, 2023.

[295] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023.

[296] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[297] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://api.semanticscholar.org/CorpusID:248512473.

[298] Megan Richards, Polina Kirichenko, Diane Bouchacourt, and Mark Ibrahim. Does progress on object recognition benchmarks improve real-world generalization?, 2023.

[299] Zachary Novack, Julian McAuley, Zachary Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning (ICML)*, 2023.

[300] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *ICLR*, 2023.

[301] Sarah M Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification, 2023. URL https://openreview.net/forum?id=3ly9cG9Ql9h.

[302] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[303] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[304] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*, 2022.

[305] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees G. M. Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization, 2023.

[306] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

[307] M. Jehanzeb Mirza, Leonid Karlinsky, Wei Lin, Mateusz Kozinski, Horst Possegger, Rogerio Feris, and Horst Bischof. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. (arXiv:2305.18287), May 2023. URL http://arxiv.org/abs/2305.18287. arXiv:2305.18287 [cs].

[308] Cristina Menghini, Andrew Delworth, and Stephen H. Bach. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. (arXiv:2306.01669), Jun 2023. doi: 10.48550/arXiv.2306.01669. URL http://arxiv.org/abs/2306.01669. arXiv:2306.01669 [cs].

[309] Ching-Yao Chuang, Jampani Varun, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka. Debiasing vision-language models via biased prompts. *arXiv preprint 2302.00070*, 2023.

[310] Ashish Seth, Mayur Hemani, and Chirag Agarwal. Dear: Debiasing vision-language models with additive residuals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 6820–6829. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00659. URL https://doi.org/10.1109/CVPR52729.2023.00659.

[311] Yuhui Zhang, Jeff Z. HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and rectifying vision models using language. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=D-zfUK7BR6c.

[312] Bang An, Sicheng Zhu, Michael-Andrei Panaitescu-Liess, Chaithanya Kumar Mummadi, and Furong Huang. More context, less distraction: Visual classification by inferring and conditioning on contextual attributes, 2023.

[313] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19187–19197, June 2023.

[314] Zhili Feng, Anna Bair, and J. Zico Kolter. Leveraging multiple descriptive features for robust few-shot image learning, 2023.

[315] Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts, 2023.

[316] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[317] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2015. URL https://api.semanticscholar.org/CorpusID:15870772.

[318] Vikram V. Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron B. Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. Beyond web-scraping: Crowdsourcing a geodiverse dataset. In *arXiv preprint*, 2023.

[319] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019.

[320] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

[321] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. URL https://api.semanticscholar.org/CorpusID:15193013.

[322] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013.

[323] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. URL https://api.semanticscholar.org/CorpusID:383200.

[324] Vikram V. Ramaswamy, Sunnie S. Y. Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability, 2023.

[325] Mazda Moayeri, Samyadeep Basu, Sriram Balasubramanian, Priyatham Kattakinda, Atoosa Chengini, Robert Brauneis, and Soheil Feizi. Rethinking copyright infringements in the era of text-to-image generative models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0OTVNEm9N4.

[326] Deepfloyd, Apr 2023. URL https://github.com/deep-floyd/IF.

[327] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

[328] Blake Brittain. Music labels' ai lawsuits create new copyright puzzle for us courts. *Reuters*, 2024. URL https://www.reuters.com/legal/music-labels-ai-lawsuits-create-new-copyright-puzzle-us-courts-2024-0 Accessed: 2024-9-26.

[329] Isaiah Poritz. Artist advances 'compressed copies' theory in stability ai suit. *Bloomberg Law*, 2024. URL https://news.bloomberglaw.com/litigation/artist-advance-compressed-copies-theory-in-stability-ai-suit. Accessed: 2024-9-26.

[330] Adobe. Fair act to protect artists in age of ai, September 12 2023. URL https://blog.adobe.com/en/publish/2023/09/12/fair-act-to-protect-artists-in-age-of-ai#:~:text=The%20right%20requires%20intent%20to,independent%20creation%20is%20a%20defense. Accessed: 2024-05-22.

[331] Paul Goldstein. *Goldstein on Copyright, 3rd edition*. Wolters Kluwer Legal & Regulatory U.S., 2014.

[332] Dc comics v. towle, 2015. URL https://casetext.com/case/comics-v-towle-2.

[333] Metro-goldwyn-mayer v. american honda motor co., 1995. URL https://law.justia.com/cases/federal/district-courts/FSupp/900/1287/2262232/.

[334] Klinger v. conan doyle estate, ltd., 2014. URL https://caselaw.findlaw.com/court/us-7th-circuit/1674598.html.

[335] Sony. Danjaq llc v. sony corp., 1998. Relating to the rights to James Bond films based on *Thunderball*.

[336] Benjamin Sobel. Elements of style: Copyright, similarity, and generative ai. *Harvard Journal of Law & Technology, Forthcoming Vol. 38, Cornell Legal Studies Research Paper Forthcoming*, 2024. Available at SSRN: https://ssrn.com/abstract=4832872 or http://dx.doi.org/10.2139/ssrn.4832872.

[337] Seán M. O'Connor. Ai replication of musical styles points the way to an exclusive rights regime. In Ryan Abbott, editor, *Research Handbook on Intellectual Property and Artificial Intelligence*. Edward Elgar, 2022. Available at SSRN: https://ssrn.com/abstract=4066915 or http://dx.doi.org/10.2139/ssrn.4066915.

[338] Tufenkian import / export ventures inc. v. einstein moomjy inc., 2003. URL https://caselaw.findlaw.com/us-2nd-circuit/1455682.html.

[339] Oren Bracha. The work of copyright in the age of machine production. *U of Texas Law, Legal Studies Research Paper*, 2023. Available at SSRN: https://ssrn.com/abstract=4581738 or http://dx.doi.org/10.2139/ssrn.4581738.

[340] Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, and Jiliang Tang. Copyright protection in generative ai: A technical perspective, 2024.

[341] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N. Metaxas, and Shiqing Ma. DIAGNOSIS: Detecting unauthorized data usages in text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=f8S3aLm0Vp.

[342] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, Yue Xing, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models, 2023.

[343] Yingqian Cui, Jie Ren, Yuping Lin, Han Xu, Pengfei He, Yue Xing, Wenqi Fan, Hui Liu, and Jiliang Tang. FT-SHIELD: A watermark against unauthorized fine-tuning in text-to-image diffusion models, 2024. URL https://openreview.net/forum?id=OQccFglTb5.

[344] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models, 2023.

[345] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion based mimicry through score distillation, 2024.

[346] Zhengyue Zhao, Jinhao Duan, Kaidi Xu, Chenan Wang, Rui Zhangp Zidong Dup Qi Guo, and Xing Hu. Can protective perturbation safeguard personal data from being exploited by stable diffusion?, 2023.

[347] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023.

[348] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models, 2022.

[349] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 47783–47803. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9521b6e7f33e039e7d92e23f5e37bbf4-Paper-Conference.pdf.

[350] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models, 2023.

[351] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models, 2023.

[352] Samyadeep Basu, Nanxuan Zhao, Vlad Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models, 2023.

[353] Generative artificial intelligence and copyright law, Sep 2023. URL https://crsreports.congress.gov/product/pdf/LSB/LSB10922.

[354] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection, 2022.

[355] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. URL https://arxiv.org/abs/2104.14294.

[356] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Milan, Italy, 2024. Springer.

[357] Wei Ren Tan, Chee Seng Chan, Hernán E. Aguirre, and Kiyoshi Tanaka. Artgan: Artwork synthesis with conditional categorial gans. *CoRR*, abs/1702.03410, 2017. URL http://arxiv.org/abs/1702.03410.

[358] Sergey Karayev, Aaron Hertzmann, Matthew Trentacoste, Helen Han, Holger Winnemoeller, Aseem Agarwala, and Trevor Darrell. Recognizing image style. In *Proceedings of the British Machine Vision Conference 2014*, BMVC 2014. British Machine Vision Association, 2014. doi: 10.5244/c.28.122. URL http://dx.doi.org/10.5244/C.28.122.

[359] C. Richard Johnson, Ella Hendriks, Igor J. Berezhnoy, Eugene Brevdo, Shannon M. Hughes, Ingrid Daubechies, Jia Li, Eric Postma, and James Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4):37–48, 2008. doi: 10.1109/MSP.2008.923513.

[360] Nanne van Noord, Ella Hendriks, and Eric O. Postma. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32:46–54, 2015. URL https://api.semanticscholar.org/CorpusID:15774940.

[361] A. J. K. Robinson. The evolution of copyright, 1476-1776. *Cambrian Law Review*, 22:55, 1991.

[362] Joseph Loewenstein. *The Author's Due: Printing and the Prehistory of Copyright*. University of Chicago Press, 2010.

[363] The statute of anne, 8 anne, c. 19, 1710. URL https://www.copyrighthistory.org/cam/tools/request/showRecord.php?id=record_uk_1710. April 10, 1710.

[364] Copyright act of 1790, 1st congress, 2nd session, chapter xv, 1790. URL https://www.copyrighthistory.org/cam/tools/request/showRecord.php?id=record_us_1790. Stat. 124-126, May 31, 1790.

[365] Copyright act of 1870, act of july 8, 1870, chapter 230, 1870. URL https://www.copyrighthistory.org/cam/tools/request/showRecord.php?id=record_us_1870. 16 Stat. 198, July 8, 1870.

[366] Robert Brauneis. Understanding copyright's first encounter with the fine arts: A look at the legislative history of the copyright act of 1870. *Case Western Reserve Law Review*, 71 (1), 2020.

[367] US Government. Act of august 24, 1912, chapter 356, 1912. URL https://www.loc.gov/law/help/statutes-at-large/62nd-congress/session-2/c62s2ch356.pdf. 37 Stat. 488, Public Law 303.

[368] US Government. The sound recording act of 1971, public law 92-140, section 3, 1971. URL https://www.govinfo.gov/content/pkg/STATUTE-85/pdf/STATUTE-85-Pg391.pdf. 85 Stat. 391.

[369] US Government. 7 u.s.c. § 114(b). United States Code, Title 7, Section 114(b).

[370] Jonathan Lopez. *The Man Who Made Vermeers: Unvarnishing the Legend of Master Forger Han van Meegeren*. St. Martin's Press, 2008.

[371] Arne Birkenstock. Beltracchi - die kunst der fälschung, 2014. Motion Picture.

[372] Shaun Greenhalgh. *A Forger's Tale: Confessions of the Bolton Forger*. The History Press, 2015.

[373] Giacomo. Generative ai, art authentication and forgery, September 2023. URL https://www.hephaestusanalytical.com/blog/generative-ai-art-authentication-and-forgery. Hephaestus Analytical.

[374] Joe Coscarelli. An a.i. hit of fake 'drake' and 'the weeknd' rattles the music world. *New York Times*, April 2023. URL https://www.nytimes.com/2023/04/19/arts/music/ai-drake-the-weeknd-fake.html.

[375] Metro-Goldwyn-Mayer v. American Honda Motor Co. 900 f. supp. 1287, 1995. U.S. District Court for the Central District of California.

[376] DC Comics v. Towle. 802 f.3d 1012, 2015. U.S. Court of Appeals for the Ninth Circuit.